

# Adaptive Extended Kalman Filter Based Indoor Positioning Algorithm Integrating Wi-Fi and PDR

Xin-Peng Zheng, Lie-Ping Zhang, Zhen-Tao Yu, and Cui Zhang\*

**Abstract**—With the rapid development of indoor positioning and navigation technologies, pedestrian indoor positioning has become a hotspot in research and applications. However, Wi-Fi fingerprint positioning is susceptible to environmental changes, leading to unstable positioning accuracy, while Pedestrian Dead Reckoning (PDR) suffers from cumulative errors. To address these issues, this paper proposes an Adaptive Extended Kalman Filter (AEKF) based indoor positioning algorithm integrating Wi-Fi and PDR. Firstly, a K-Nearest Neighbors fingerprint matching method based on position range constraints is designed to enhance the continuity of position estimation and reduce positioning errors. Secondly, in the PDR component, adaptive peak detection, improved Weinberg step length estimation, and quaternion-based heading estimation are introduced to reduce cumulative errors. Finally, in the fusion stage, an AEKF method is proposed, which employs an Adaptive Dung Beetle Optimization (ADBO) algorithm to dynamically adjust the covariance matrix of the Extended Kalman Filter. Additionally, an adaptive weight adjustment mechanism is introduced to dynamically assign observation weights based on the quality of Wi-Fi and PDR observations, thereby improving the positioning accuracy and robustness of the fusion positioning system. Experimental results show that the proposed method achieves an average positioning error of 0.627 m, representing a 19.7% to 66.0% reduction compared to single positioning algorithms and other fusion methods. It also demonstrates faster convergence and higher positioning stability in the cumulative distribution function curve, providing a high-accuracy and high-robustness fusion positioning solution for indoor positioning systems.

**Index Terms**—indoor positioning, Adaptive Extended Kalman Filter, Wi-Fi fingerprint positioning, Pedestrian Dead Reckoning, Adaptive Dung Beetle Optimization.

## I. INTRODUCTION

WITH the rapid proliferation of mobile devices, location-based services have gradually integrated into

various aspects of daily life. Pedestrian navigation systems typically rely on Global Positioning System signal (GPS) in outdoor environments. However, in complex indoor environments such as airports, railway stations, and shopping malls, traditional GPS struggle to provide accurate, continuous, and effective positioning due to signal obstruction from buildings and limitations in the power of navigation chips [1]. To achieve reliable indoor positioning, indoor positioning technologies have been extensively developed and applied in recent years. Common technologies include Wi-Fi [2], Ultra-Wideband (UWB) [3], Zigbee [4], and Bluetooth [5], and others. Among various indoor positioning technologies, Wi-Fi and Pedestrian Dead Reckoning (PDR) positioning methods have emerged as the most practical and easily deployable solutions, owing to the widespread adoption of smartphones and WLAN. Wi-Fi positioning has gradually become a mainstream indoor positioning technique due to its low cost, wide deployment, and relatively high positioning accuracy. Among them, Wi-Fi fingerprinting method based on Received Signal Strength Indication (RSSI) offers advantages such as ease of implementation, low cost, and no requirement to measure the coordinates of Access Points (APs) or the angle and distance between receiver and AP [6]. However, Wi-Fi signal is easily affected by environmental interference, leading to significant fluctuations in positioning accuracy. PDR, by contrast, estimates a pedestrian's real-time step length and heading using data from accelerometers, gyroscopes, and magnetometers in Inertial Measurement Unit (IMU), and calculates current position through cumulative integration [7]. Since this approach does not rely on external signals, it enables standalone positioning. However, since position is calculated recursively, the positioning error increases progressively over time. Since Wi-Fi positioning and PDR each offer distinct advantages and can both be implemented on consumer-grade portable devices, their integration presents a promising solution. Wi-Fi positioning can provide absolute position references to correct PDR's cumulative errors, while PDR can compensate for the limited update rate of Wi-Fi in dynamic environments. Therefore, fusing of Wi-Fi and PDR is considered an effective approach to achieving high-accuracy, stable, and real-time indoor positioning, with significant research value and application potential.

To overcome the limitations of single positioning technologies in terms of accuracy and stability, researchers have conducted extensive studies on Wi-Fi and PDR fusion methods and proposed various optimization strategies to

Manuscript received March 2, 2025; revised June 24, 2025.

This work was supported in part by National Natural Science Foundation of China No.61741303; Guangxi Natural Science Foundation Project No. 2025GXNSFAA069942; Guangxi Key Laboratory of Spatial Information and Geomatics (Guilin University of Technology) No.21-23821-16; and Innovation Project of Guangxi Graduate Education No.YCSW2025411.

Xin-Peng Zheng is a postgraduate student at Key Laboratory of Advanced Manufacturing and Automation Technology, Guilin University of Technology, Guilin 541006, China (e-mail: 2120231396@glut.edu.cn).

Lie-Ping Zhang is a professor at Guangxi Key Laboratory of Special Engineering Equipment and Control, Guilin University of Aerospace Technology, Guilin 541004, China (e-mail: zlp@guat.edu.cn).

Zhen-Tao Yu is a postgraduate student at Guilin University of Technology, Guilin 541006, China (e-mail: yyyuzhentao@163.com).

Cui Zhang is an associate professor at School of Information Engineering, Nanning College of Technology, Guilin 541006, China (corresponding author, phone: +086-181-7268-6917; e-mail: 18172686917@163.com).

enhance system performance. Zhu et al. [8] proposed an indoor navigation method based on the fusion of PDR and Wi-Fi, in which the gyroscope output from the PDR is used as the primary observation. Wi-Fi is then employed to correct the positioning, and an Extended Kalman Filter (EKF) is finally used to fuse the PDR and Wi-Fi, resulting in more reliable and accurate positioning. Qian et al. [9] proposed an improved particle filter-based indoor Wi-Fi/PDR positioning. The method improves Wi-Fi positioning accuracy through sub-region division and K-Nearest Neighbor (KNN), and integrates Wi-Fi and PDR using a particle filter, with resampling optimized by a clonal selection algorithm. This approach effectively improves positioning accuracy. Mehrabian et al. [10] proposed an indoor positioning system through improved RSSI and PDR methods. Firstly, a weight-based optimization filter is used to refine raw RSSI values. Then, the RSSI data is processed using KNN, a logarithmic path loss model, and a weighted centroid approach. Finally, Kalman Filtering (KF) is applied to fuse RSSI and PDR, further enhancing positioning accuracy. Liu et al. [11] proposed an enhanced Wi-Fi/PDR indoor positioning algorithm. A particle swarm optimization algorithm is used to optimize the weighted KNN for Wi-Fi positioning, and the artemisinin optimization algorithm is applied to improve the performance of particle filter-based fusion, thereby enhancing the effectiveness of Wi-Fi/PDR integration. Chen et al. [12] proposed a PDR/Wi-Fi indoor navigation algorithm based on a federated particle filter. In this method, PDR and heading information are used in the state transition model, while step length and Wi-Fi matching results are used in the observation model. The federated particle filter is employed to fuse PDR and Wi-Fi, effectively reducing positioning errors. Yu et al. [13] proposed a neural network-based Wi-Fi/PDR fusion positioning. A long short-term memory network is used to learn historical pedestrian motion patterns to reduce cumulative errors in PDR. Finally, a backpropagation (BP) neural network is applied to enhance the performance of the EKF for fusing Wi-Fi and PDR, thereby improving positioning accuracy. Xu et al. [14] proposed a collaborative adaptive multi-source data fusion method for smartphones. In the offline stage, Wi-Fi fingerprint data is trained using a Multi-Layer Perceptron (MLP). In the online stage, real-time Wi-Fi fingerprints are matched, and step length and heading are integrated into the PDR to estimate position. Finally, the EKF is used to dynamically adjust the weights of individual positioning algorithms, further improving positioning accuracy.

The methods mentioned above improve the accuracy and stability of indoor positioning by integrating Wi-Fi and PDR. However, current Wi-Fi/PDR fusion approaches still face the following challenges: 1. Traditional Wi-Fi fingerprinting methods often ignore the spatial continuity between positions, resulting in large positioning errors. 2. PDR methods are sensitive to noise, leading to inaccurate step count estimation. In addition, models for estimating step length often lack adaptability to gait variations and complex environments, resulting in increased cumulative errors. 3. Conventional EKF methods assume a fixed covariance matrix, neglecting the dynamic variations of indoor environments and the time-varying nature of noise. Also, when combining Wi-Fi

and PDR, the importance of each sensor's data isn't changed over time, which can lead to problems if one sensor stops working well, making the system less flexible in complicated situations. Therefore, we propose an Adaptive Extended Kalman Filter (AEKF) based indoor positioning algorithm integrating Wi-Fi and PDR, aiming to enhance the adaptability and positioning accuracy of the fusion algorithm under varying environmental conditions. The main contributions are summarized as follows:

(1) KNN Fingerprint Matching Method Based on Position Range Constraints: To address the positioning errors caused by the lack of continuity between positions in traditional Wi-Fi fingerprinting, this paper proposes a KNN fingerprint matching method based on position range constraints. In the offline stage, RSSI values are sorted and intersected, followed by Gaussian filtering to extract stable signal features and construct a reliable fingerprint database. In the online stage, the proposed method utilizes information from the previous known position to constrain the KNN matching range, thereby enhancing spatial continuity between estimated positions. The method effectively reduces positioning errors and improves the accuracy and stability of Wi-Fi positioning.

(2) Improved PDR Positioning: To reduce cumulative errors in PDR, an adaptive peak detection method is employed for step detection to eliminate interference from environmental noise and sensor drift, thereby improving the accuracy of step counting. For step length estimation, the Weinberg model is improved by introducing acceleration variance, enhancing the precision of step length calculation. Additionally, a quaternion-based method is adopted for heading estimation, which avoids the gimbal lock problem associated with Euler angles and reduces computational complexity, resulting in more stable and accurate heading estimation.

(3) Adaptive Dung Beetle Optimization Algorithm (ADBO): To address the limitations of the original Dung Beetle Optimization (DBO) algorithm in terms of global search capability, convergence accuracy, and population diversity, this paper proposes an improved ADBO algorithm. Firstly, population diversity is enhanced by introducing a good point set strategy. Secondly, the rolling behavior is improved using the multiplication strategy from the Arithmetic Optimization Algorithm (AOA), which strengthens global search capability and convergence speed. Thirdly, the foraging behavior is optimized by adopting the joiner position strategy from the Sparrow Search Algorithm (SSA), guiding the population closer to the optimal solution. Finally, an adaptive t-distribution mutation perturbation is introduced to avoid local optima. Experiments on 13 CEC2005 benchmark functions demonstrate that ADBO achieves superior performance compared to five other algorithms, particularly in convergence accuracy, speed, and global search ability and robustness, demonstrating its effectiveness for complex optimization problems.

(4) AEKF-Based Fusion Positioning Method: In the fusion stage of Wi-Fi and PDR, an AEKF-based fusion positioning method is proposed. By introducing ADBO to dynamically adjust the covariance matrix of the EKF, the filter is made adaptive to dynamic environmental changes. Additionally, an adaptive weight adjustment mechanism is introduced to

dynamically assign observation weights based on the quality of Wi-Fi and PDR measurements, enhancing the system's fault tolerance to sensor errors. Experimental results in typical indoor environments demonstrate that the proposed AEKF-based fusion positioning method significantly outperforms both single-source positioning algorithms and other fusion approaches in terms of positioning accuracy, robustness, and adaptability to environmental changes.

This paper is organized in the following manner: Section II introduces the KNN fingerprint matching method based on position range constraints. Section III presents the improved PDR positioning method. Section IV describes the AEKF-based fusion positioning method. Section V discusses the experimental validation and result analysis. Finally, Section VI concludes the paper.

## II. KNN FINGERPRINT MATCHING METHOD BASED ON POSITION RANGE CONSTRAINTS

Wi-Fi fingerprinting positioning typically uses RSSI as fingerprint features and consists of two stages, as illustrated in Fig. 1. In the offline stage, reference points (RPs) are selected within the target area, and RSSI data is collected to establish a mapping between fingerprints and their corresponding position coordinates, thereby constructing the fingerprint database. In the online stage, a fingerprint matching algorithm compares the fingerprint of the test point with those in the database to determine the current position [15]. The KNN algorithm is widely used for fingerprint matching due to its low computational complexity and lack of training requirements [16].

However, in fingerprint positioning, the RSSI signals collected by receiver vary continuously across different walking areas, leading to the following two issues: 1. In the offline stage, due to fluctuations in signal strength caused by sampling time, environmental noise, and obstacles, the resulting fingerprint database often lacks stability and robustness; 2. In the online matching stage, traditional direct positioning methods typically ignore the continuity between adjacent positions, thereby affecting positioning accuracy. To address these issues, in the offline stage, RSSI signals are sorted and filtered using Gaussian smoothing after intersection operations, to extract fingerprint data that best represent the unique characteristics of the target position. In the online stage, a KNN fingerprint matching method based on position range constraints is adopted, which effectively reduce positioning errors to a certain extent. The specific steps are as follows:

Step 1: In the offline phase, according to the size and spatial distribution of the target area,  $i$ -th RP and  $j$ -th AP are deployed, with each AP distinguished by its MAC addresses. For the  $i$ -th RP, a fingerprint entry is generated by collecting the RSSI from the  $j$ -th AP. After completing data collection at all RPs, the offline fingerprint database is constructed.

$$FM = \begin{bmatrix} x_1^{RP} & y_1^{RP} & RSSI_{1,1} & \cdots & RSSI_{1,j} \\ x_2^{RP} & y_2^{RP} & RSSI_{2,1} & \cdots & RSSI_{2,j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_i^{RP} & y_i^{RP} & RSSI_{i,1} & \cdots & RSSI_{i,j} \end{bmatrix} \quad (1)$$

where  $j$  is number of APs, and  $RSSI_{i,j}$  represents the RSSI value of the  $j$ -th AP at the  $i$ -th RP.

Step 2: First, for each RP, the RSSI list is sorted in descending order of signal strength. The top  $M$  strongest RSSI values are selected as features to ensure the representativeness of the fingerprint database. The filtered RSSI feature vector for the  $i$ -th RP is given by

$$RSSI_i^{sorted} = (RSSI_{i,1}, RSSI_{i,2}, \dots, RSSI_{i,M}) \quad (2)$$

Step 3: Next, intersection operations are performed on multiple sampling results for each RP to extract signals that consistently appear across all samples. This process helps identify the most stable signal features. The intersection set of the sampling data for the  $i$ -th RP is calculated as:

$$RSSI_i^{stable} = \bigcap_{i=1}^T RSSI_i^{sorted}(t) \quad (3)$$

where  $T$  represents the number of signal samples in the RSSI data.

Step 4: The RSSI values at the same position tend to follow a normal distribution. Therefore, Gaussian filtering can be applied to process each set of RSSI data to extract the RSSI value with the highest probability. This effectively reduces the impact of low-probability and strong interference noise on the collected RSSI values. The RSSI for the  $i$ -th RP, assumed to follow a normal distribution, can be expressed as:

$$RSSI_i^{stable} \sim N(\mu, \sigma^2) \quad (4)$$

The probability distribution function of signal strength can be represented as:

$$f(RSSI_i^{stable}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(RSSI_i^{stable} - \mu)^2}{2\sigma^2}} \quad (5)$$

where  $\mu$  and  $\sigma$  represent the mean and standard deviation, respectively, and the expression is as follows:

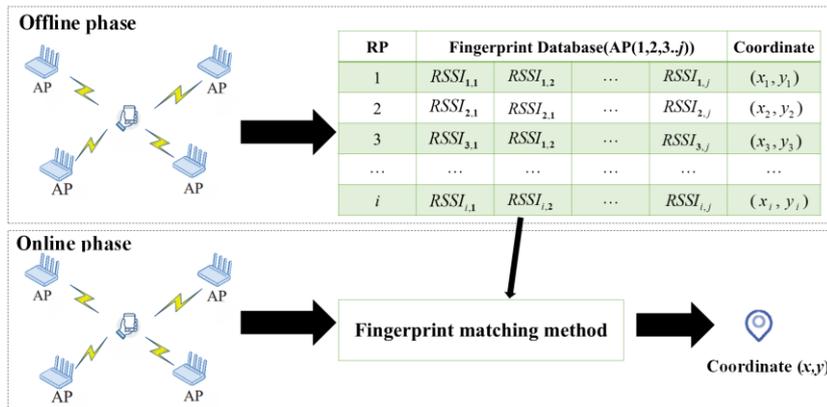


Fig. 1. Wi-Fi fingerprint positioning principle.

$$\mu = \frac{1}{L} \sum_{i=1}^L RSSI_i^{stable} \quad (6)$$

$$\sigma = \sqrt{\left( \frac{1}{L-1} \sum_{i=1}^L (RSSI_i^{stable} - \mu)^2 \right)} \quad (7)$$

where  $L$  represents the number of RSSI samples after performing the intersection operation. By transforming the formula, it can be normalized to follow a standard normal distribution, as follows:

$$\frac{RSSI_i^{stable} - \mu}{\sigma} \sim N(0,1) \quad (8)$$

The RSSI values with a probability of occurrence within 90% are selected. By referring to standard normal distribution table, the value range can be obtained as:

$$\left| \frac{RSSI_i^{stable} - \mu}{\sigma} \right| < 1.65 \quad (9)$$

Through the above steps, the RSSI values of all RPs within the range  $(\mu - 1.65\sigma, \mu + 1.65\sigma)$  are used to construct offline fingerprint database.

Step 5: Online Phase: To enhance the correlation between each position point, a KNN fingerprint matching method based on position range constraints is used. When estimating the current position, the user's previously known position is utilized to strengthen the association between adjacent positioning points, thereby reducing positioning error. Firstly, the Euclidean distance between the  $i$ -th RP and the target positioning point is calculated as:

$$d_{l,i} = \sqrt{\sum_{j=1}^n (RSSI_{l,j} - RSSI_{i,j})^2} \quad (10)$$

where  $RSSI_{l,j}$  is RSSI of the  $j$ -th AP at the target positioning point.

Step 6: In the process of continuous signal-based positioning, the movement speed of the target is limited. Therefore, within a short time interval, the target is not to move beyond a certain specific range. Based on this characteristic, the target's previous location is recorded, and an elliptical area is drawn centered on that position to constrain the possible range of the target at the next time step. This method effectively captures the movement trend of the target, prioritizes RPs in the primary direction, and reduces the impact of lateral errors, thereby improving positioning accuracy and stability. To achieve this, the distance between each RP  $i$  and the previous location  $(x_{pre}, y_{pre})$  is used to compute a weight, assigning higher weights to RPs closer to the previous position, while weights for distant RPs rapidly approach zero. The weight is calculated as follows:

$$W_{l,i} = \exp\left( \frac{(x_i - x_{pre})^2}{4d_1^2} + \frac{(y_i - y_{pre})^2}{4d_2^2} \right) \quad (11)$$

where  $W_{l,i}$  is weight of the  $i$ -th RP for the target point  $l$ ;  $d_1$  is the major axis of the ellipse, aligned with the primary direction of the target's movement. The major axis is relatively long to cover the main forward movement range of the target, reflecting its inertial motion characteristics.  $d_2$  is the minor axis of the ellipse, perpendicular to the main motion direction of the target. The length of the minor axis is smaller, limiting lateral displacement and ensuring that the weights of lateral RPs decay quickly, thereby reducing unnecessary lateral errors. When the target turns or changes its direction of movement, the lengths and orientations of the major and minor axes can be dynamically adjusted. If the new movement direction is opposite to the previous one, the roles of the major and minor axes are switched—that is, the previous minor axis becomes longer and the major axis becomes shorter. This adaptation allows the system to better follow the target's movement changes, thus improving both positioning accuracy and reliability. The values of  $d_1$  and  $d_2$  are determined by the target's actual moving speed during the positioning and the frequency of online positioning.

The distance between the  $i$ -th RP and the target point is given by:

$$\bar{d}_{l,i} = \frac{W_{l,i} \times d_{l,i}}{\sum_{i=1}^M W_{l,i}} \quad (12)$$

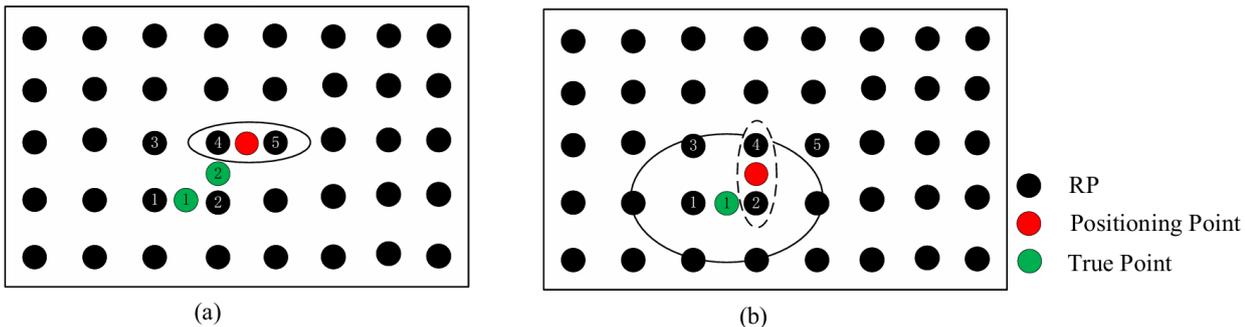
where  $M$  is the total number of RPs.

Step 7: Select the  $K$  RP closest to the target point and estimate the position based on the weighted results. The final target  $(x_l, y_l)$  position is given by:

$$\omega_{l,i} = \frac{1/\bar{d}_{l,i}}{\sum_{k=1}^K 1/\bar{d}_{l,i}} \quad (13)$$

$$(x_l, y_l) = \sum_{i=1}^K \omega_{l,i} (x_i, y_i) \quad (14)$$

Fig. 2 illustrates the schematic diagrams of KNN-based positioning under conventional conditions and with position range constraints, respectively. In both subfigures, True Point 1 represents the target's previous position, while True Point 2 indicates its current position. The value of  $K$  is set to 2. As shown in Fig. 2 (a), the conventional KNN-based positioning method tends to produce estimates that deviate significantly



(a) Conventional KNN Positioning Matching Method

(b) KNN Fingerprint Matching Method Based on Position Range Constraints

Fig. 2. KNN-based positioning under conventional conditions and with position range constraints.

from the actual position. In contrast, Fig 2. (b) demonstrates the proposed position range-constrained method, where an elliptical region is centered at True Point 1, with predefined radii for the major and minor axes. Reference Point 5, which lies outside this region, is excluded from the candidate set. As a result, the algorithm selects candidate points only within a reasonable spatial range. The final estimated position under position range constraints is closer to the true current location, effectively reducing positioning errors.

### III. IMPROVED PDR POSITIONING

PDR utilizes data from built-in smartphone sensors such as accelerometer, gyroscope, and magnetometer. By combining these sensor readings with the pedestrian's known current position, the PDR algorithm can estimate the number of steps, step length, and heading direction to achieve localization [17]. The principle of PDR positioning is illustrated in Fig. 3. Suppose the pedestrian's position at time  $t-1$  is  $(x_{t-1}, y_{t-1})$ , and during time from  $t-1$  to  $t$  the pedestrian moves a distance of  $l_t$  in the direction of heading angle  $\theta_t$ , then pedestrian's position at time  $t+1$ :

$$\begin{cases} x_t = x_{t-1} + l_t \sin \theta_t \\ y_t = y_{t-1} + l_t \cos \theta_t \end{cases} \quad (15)$$

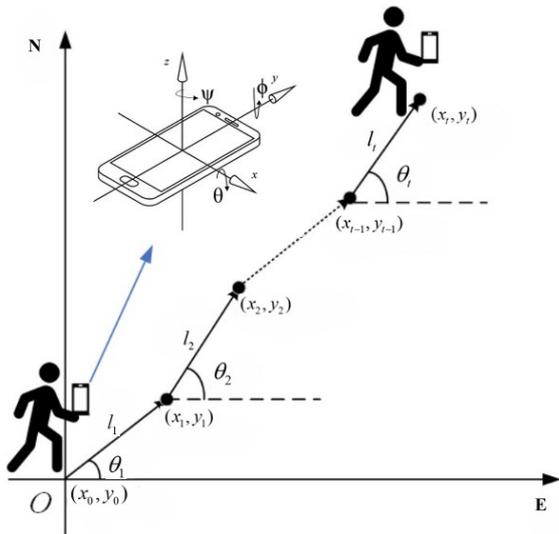


Fig. 3. Principle of PDR Positioning.

#### A. Step Count Estimation Using Adaptive Peak Detection

In step detection algorithms, methods such as peak detection, autocorrelation, and zero-crossing detection offer relatively high and stable accuracy in step counting [18]. Considering its low computational complexity, this study adopts the peak detection method for step estimation [19]. The peak detection method leverages the quasi-sinusoidal nature of acceleration signals during human walking. A step is registered each time a peak is detected, and the corresponding timestamp is recorded. However, due to external interference and system noise, the output signal may contain spikes or false peaks.

To address this issue, an adaptive peak detection-based step estimation method is proposed to suppress noise and improve step detection accuracy. The step detection process is as follows:

Step 1: In step detection, the vertical acceleration needs to be calculated first. This method removes the gravity component from the accelerometer data to extract vertical acceleration for step detection. The angle  $\theta$  is calculated to convert the gravity acceleration component from the smartphone's coordinate system to the vertical acceleration component in the Earth coordinate system:

$$\theta = \arctan\left(\frac{|g_z|}{|g_y|}\right) \quad (16)$$

where  $g_z$  and  $g_y$  are the  $z$ -axis and  $y$ -axis components of gravity acceleration, respectively. The vertical acceleration is then calculated as:

$$a_{\text{vertical}} = a_y \cdot \cos(\theta) + a_z \cdot \sin(\theta) \quad (17)$$

where  $a_z$  and  $a_y$  are the  $z$ -axis and  $y$ -axis acceleration components in the smartphone coordinate system, respectively.

Step 2: Moving average filtering is a simple and effective smoothing method. It replaces the current data point with the average of several preceding and succeeding data points to reduce fluctuations caused by noise. A window size of 5 is set to calculate the moving average. The moving average value at the  $i$ -th point is calculated as:

$$a_{\text{vertical smooth}}[i] = \frac{1}{5} \sum_{j=i-2}^{i+2} a_{\text{vertical}}[j] \quad (18)$$

Step 3: Set a sliding window, which is used for segmenting and detecting step peaks. The sliding window is calculated as follows:

$$\text{slide} = 40 \times \frac{f}{100} \quad (19)$$

where  $f$  is the sampling frequency, which is set to 100 Hz. Thus, the sliding window records a peak value every 40 data points.

Step 4: Acceleration lower than  $0.2 \times g$  typically indicates minor movements, while acceleration higher than  $2 \times g$  usually represents intense motion or sudden shocks. By setting the acceleration range to  $0.2 \times g$  to  $2 \times g$ , data resulting from minor or intense movements are excluded, allowing for more accurate capture of acceleration peaks during normal walking.

$$0.2 \times g \leq a_{\text{vertical smooth}} \leq 2 \times g \quad (20)$$

where  $g = 9.794 \text{ m/s}^2$  is the gravitational acceleration. A peak is detected every slide data points, and the qualified peaks are recorded as steps.

Step 5: It is crucial to prevent the detection of multiple peaks within a short period, as these peaks may represent false steps caused by device vibrations or slight movements. Setting a time interval between two peaks effectively avoids detecting multiple peaks in a short time, thereby improving the accuracy of step count detection. In standard walking patterns, the normal walking frequency ranges from 0.5 Hz to 2.5 Hz, meaning each step lasts about 0.4s to 2s. First, calculate the time interval between adjacent step points:

$$\Delta t = t_{\text{current}} - t_{\text{previous}} \quad (21)$$

where  $t_{\text{current}}$  is the timestamp of the current step point, and  $t_{\text{previous}}$  is the timestamp of the previous step point. To prevent false detection, the time interval between each step is

checked to ensure it falls within the normal range. The condition is defined as follows:

$$\begin{aligned} & \text{if } \Delta t > 2 \text{ or } \Delta t < 0.4, \text{ current step is deleted} \\ & \text{if } 0.4 \leq \Delta t \leq 2, \text{ current step is retained} \end{aligned} \quad (22)$$

### B. Improved Weinberg Step Length Estimation

Commonly used step length estimation models include linear, nonlinear models, and machine learning models [20]. Among them, nonlinear models are based on the statistical characteristics of acceleration changes during walking and establish a nonlinear relationship with acceleration to estimate step length with higher accuracy. These models are adaptable to various motion states and have lower complexity compared to machine learning models [21]. Typical nonlinear models include the Weinberg, Scarlet, and Kim models. The Weinberg model is simpler and more efficient than the Scarlet and Kim models and demonstrates excellent accuracy and stability [22]. Therefore, the Weinberg model is adopted. The step length estimated by the Weinberg model is:

$$L = K \cdot \sqrt[4]{a_{\max} - a_{\min}} \quad (23)$$

where  $a_{\max}$  and  $a_{\min}$  are the maximum and minimum accelerations detected during a step, respectively, and  $K$  is a constant.

The Weinberg model typically uses the difference in acceleration amplitude to estimate step length but overlooks the rate of acceleration change and the degree of fluctuation during a step. Therefore, by introducing the acceleration variance  $\sigma_a$ , the model's adaptability to different gaits is enhanced. The acceleration variance  $\sigma_a$  is given by:

$$\sigma_a = \frac{1}{N} \sum_{i=1}^N (a_i - \bar{a})^2 \quad (24)$$

where  $a_i$  is the  $i$ -th acceleration within  $k$  steps, and  $\bar{a}$  is the mean of all accelerations over  $k$  steps. The final improved Weinberg model is:

$$L = K \cdot \sqrt[4]{a_{\max} - a_{\min}} + B \cdot \sigma_a \quad (25)$$

where  $B$  is a fitting parameter.

### C. Quaternion-Based Heading Estimation

Common heading estimation methods include the Euler angle, direction cosine matrix (DCM), and quaternion methods [23]. The Euler angle method directly solves Euler angle differential equations, making it easy to understand and computationally simple. However, it suffers from the gimbal lock problem, where one degree of freedom is lost, leading to limited accuracy in attitude estimation [24]. Compared to the Euler angle method, the DCM method avoids the singularity issue but requires solving multiple differential equations at each update, thereby increasing computational complexity. The quaternion method, on the other hand, requires solving four differential equations for attitude estimation. Although it involves slightly higher computational load than the Euler method, it offers higher precision, avoids singularities, and significantly reduces the computational burden compared to the DCM method [25].

Therefore, this paper adopts the quaternion method for pedestrian heading estimation, converting data collected by the accelerometer, gyroscope, and magnetometer into quaternion representations. The basic definition of the

quaternion method is as follows:

$$q = q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k \quad (26)$$

where  $q_0$  represents the rotation magnitude, and  $q_1, q_2, q_3$  represent the rotation axis.  $i, j, k$  are a set of unit imaginary numbers representing three mutually orthogonal unit vectors. The components of the quaternion satisfy the following relationship:

$$i^2 + j^2 + k^2 = i \cdot j \cdot k = -1 \quad (27)$$

The rotation matrix that represents the quaternion from the navigation frame to the body frame is given as follows:

$$C_n^b(q) = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (28)$$

The orientation of the smartphone relative to the navigation coordinate system, calculated using the rotation matrix given in Equation (28), is given as follows:

$$\begin{cases} \text{pitch} = \sin^{-1} \left( \frac{C_n^b}{32} \right)_{32} = \sin^{-1} 2(q_2q_3 + q_0q_1) \\ \text{roll} = \tan^{-1} \left( \frac{C_n^b}{31} \right)_{31} = \tan^{-1} \left( \frac{2(q_0q_2 - q_1q_3)}{q_0^2 - q_1^2 - q_2^2 + q_3^2} \right) \\ \text{yaw} = \tan^{-1} \left( \frac{C_n^b}{12} \right)_{12} = \tan^{-1} \left( \frac{2(q_0q_3 - q_1q_2)}{q_0^2 - q_1^2 + q_2^2 - q_3^2} \right) \end{cases} \quad (29)$$

where *pitch* represents to the pitch angle, *roll* to the roll angle, and *yaw* to the heading angle. During the experiment, the pedestrian held the smartphone horizontally in front of the chest, with the front of the device pointing straight ahead. Therefore, this paper directly uses the *yaw* angle as the heading angle.

## IV. AEKF FUSION POSITIONING

### A. ADBO

The DBO, introduced by Xue et al. in 2023, is a recent swarm intelligence method inspired by the natural behaviors of dung beetles. These behaviors—rolling, breeding, foraging, and stealing—are modeled through different beetle types: rollers, breeders, foragers, and stealers, respectively [26]. The algorithm updates solutions based on these behavioral strategies. The update process is as follows:

(1) Rolling Behavior: Dung beetles are responsible for rolling dung balls to a safe place to hide it. They use celestial cues—particularly sunlight and polarized light—for navigation to ensure the dung ball rolls along a straight path. In the search space, rolling beetles move along a specific path and update their positions accordingly. The position update of a rolling beetle can be defined as:

$$\begin{aligned} x_i(t+1) &= x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta X \\ \Delta X &= |x_i(t) - X^w| \end{aligned} \quad (30)$$

where  $t$  is the current number of iterations,  $x_i(t)$  represents the position information of the  $i$ -th beetle at the  $t$ -th iteration,  $k \in (0, 0.2]$  is a constant representing adjustment coefficient,  $b$  is a random number between (0,1),  $X^w$  represents the global optimal position,  $\Delta X$  is used to simulate the variability in the light intensity, and  $\alpha$  is a natural coefficient with a value of either 1 or -1.

When faced with obstacles, dung beetles exhibit a dancing behavior to adjust their orientation and seek an alternative route. The position adjustment during this behavior is described by the following equation:

$$x_i(t+1) = x_i(t) + \tan \theta |x_i(t) - x_i(t-1)| \quad (31)$$

where  $\theta$  represents the deflection angle, which lies within the range  $[0, \pi]$ . When  $\theta$  takes the values  $0, \pi/2, \text{ or } \pi$ , the position remains unchanged.

(2) Breeding Behavior: Female dung beetles roll dung balls to safe sites for egg-laying, creating a secure environment for their young. Based on this behavior, a boundary selection strategy is designed to simulate the egg-laying area, as shown below:

$$\begin{cases} Lb^* = \max(X^* \times (1-R), Lb) \\ Ub^* = \min(X^* \times (1+R), Ub) \end{cases} \quad (32)$$

where  $Lb^*$  and  $Ub^*$  denote the lower and upper limits of the egg-laying region,  $X^*$  stands for the current local best position,  $R = 1 - t/T_{\max}$ ,  $T_{\max}$  is the maximum number of iterations.  $Lb$  and  $Ub$  represent the lower and upper bounds of the overall optimization problem.

During each iteration, a female dung beetle lays a single egg. The boundaries of the egg-laying region are adaptively modified, primarily influenced by the parameter  $R$ . As a result, the breeding beetle's position is also updated dynamically throughout the iterations. The corresponding position update formula is given below:

$$x_i(t+1) = X^* + b_1 \times (x_i(t) - Lb^*) + b_2 \times (x_i(t) - Ub^*) \quad (33)$$

where  $b_1$  and  $b_2$  represent two independent random vectors of size  $1 \times D$ , and  $D$  represents the dimension of the optimization problem.

(3) Foraging Behavior: After hatching, young beetles emerge from the ground and begin searching for food. To guide this behavior, an optimal foraging area is defined, helping them locate food more efficiently. The specific formula used to determine this foraging region is presented below:

$$\begin{cases} Lb^b = \max(X^b \times (1-R), Lb) \\ Ub^b = \max(X^b \times (1+R), Ub) \end{cases} \quad (34)$$

where  $X^b$  represents the global best position,  $Lb^b$  and  $Ub^b$  denote the lower and upper boundaries of the optimal foraging region, respectively. Once the optimal foraging area is determined. The specific formula used to determine this foraging region is presented below:

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b) \quad (35)$$

where  $C_1$  is a random number following a normal distribution, and  $C_2$  is a random vector within the range  $(0, 1)$ .

(4) Stealing Behavior: Stealing dung beetles steal dung balls from other beetles. They do not steal from random locations but instead target optimal stealing spots, which are areas near the global best position. The position update of a stealing beetle is given by:

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^b| + |x_i(t) - X^b|) \quad (36)$$

where  $x_i(t)$  represents the position of the  $i$ -th stealing beetle in the  $t$ -th iteration,  $g$  is a random vector of size  $1 \times D$  following a normal distribution, and  $S$  is a constant.

Although DBO has significant advantages in optimization performance and convergence efficiency, it suffers from

several limitations, including low population diversity, insufficient global search ability, low convergence accuracy, and a tendency to fall into local optima. These shortcomings lead to inadequate global exploration and local exploitation, leaving considerable room for improvement in the overall algorithm. To address these limitations, this paper proposes an ADBO. First, a good point set strategy is employed to enhance population diversity. Second, the rolling behavior is improved using the multiplication operation strategy from AOA to enhance global search ability and convergence speed. Third, the foraging behavior is enhanced by integrating the joiner position update strategy from SSA, which helps guide the population toward the global optimum and improves both convergence speed and accuracy. Finally, in the later stage of iteration, an adaptive t-distribution mutation is applied to prevent the algorithm from falling into local optima. These four strategies are designed to further enhance the overall performance of the DBO algorithm.

#### 1) Good point set for Population Initialization

Traditional DBO uses the rand function for population initialization, which introduces high randomness and uncertainty. This often leads to population clustering in the early stages, weakening global exploration and causing premature convergence to local optima. In contrast, the good point set method can uniformly cover the search space with fewer sample points. Compared with other uniform distribution strategies (such as chaotic mapping and quasi-Monte Carlo methods), which rely on randomness and often result in uneven sampling, the good point set method provides more stable and uniform coverage. To improve the algorithm's global exploration ability in the early stages of optimization and maintain a uniform population distribution throughout the solution space, this study employs the good point set method for initializing the population. Its basic principle is as follows:

In a  $s$ -dimensional space, consider a unit cube  $G_s$ , where the point set exists as follows:

$$P_n(k) = \left\{ \left\{ \{r_1^{(n)}k\}, \{r_2^{(n)}k\}, \dots, \{r_s^{(n)}k\} \right\}; k=1, 2, \dots, n \right\} \quad (37)$$

The discrepancy of the point set satisfies  $\varphi(n) = C(r, \varepsilon)n^{-1+\varepsilon}$ , where  $C(r, \varepsilon)$  is a constant dependent only on  $r$  and  $\varepsilon$  (any positive number). Let  $P_n(k)$  be the good-point set, and  $r$  be the good point. The formula for calculating the point  $r$  is as follows:

$$r_i = 2 \cos \frac{2\pi i}{p}, 1 \leq i \leq s \quad (38)$$

where  $p$  is the smallest prime number satisfying  $(p-3)/2 \geq s$ .

By using the good-point-set initialization, the formula used for the updated population initialization is presented below:

$$x_i(j) = (Ub_j - Lb_j) \cdot \{r_j^{(i)} \cdot k\} + Lb_j \quad (39)$$

where  $Ub_j$  and  $Lb_j$  represent the lower and upper bounds of the  $j$ -th dimension, respectively.

#### 2) Rolling Behavior with Multiplication Operator Strategy

To enhance the global search capability of DBO, this paper replaces the traditional rolling behavior with the multiplication operator from the AOA. By introducing this operator [27], the position update process achieves more efficient global distribution, improving the algorithm's exploration ability and flexibility during the search. Compared with the original rolling behavior, the

multiplication and division operators can perform finer scaling and distribution adjustments on the individual's position, thus improving the global search efficiency while maintaining diverse exploration, and effectively avoiding local optima. The position update formula for the rolling behavior is as follows:

$$x_i(t) = \begin{cases} X^b \cdot MOP \cdot ((Ub - Lb) \cdot \mu + Lb), \delta < m \\ x_i(t) + \tan \theta |x_i(t) - x_i(t-1)|, \delta \geq m \end{cases} \quad (40)$$

where  $\delta = \text{rand}(1)$ ,  $m \in (0.5, 1]$ ,  $\mu$  is the adjustment control parameter, set to 0.499 in this paper. The formula for calculating the Math Optimizer Probability (MOP) is as follows:

$$MOP(t+1) = 1 - \left(\frac{t}{T_{\max}}\right)^\alpha \quad (41)$$

where  $\alpha$  is a sensitivity parameter, with a value of 5 used in this paper.

### 3) Foraging Behavior with SSA Followers Strategy

When small dung beetles move toward the global best position, their foraging update strategy is relatively simple and highly random, resulting in limited convergence speed and accuracy. Inspired by the joiners in SSA, which follow the discoverers for foraging, this update strategy is introduced into the foraging process of dung beetles to enhance the algorithm's convergence speed and accuracy. The position update formula for small beetles during foraging is as follows:

$$x_i(t+1) = \begin{cases} Q \times \exp\left(\frac{X^w - x_i(t)}{t^2}\right), i > \frac{pop_3}{2} \\ X^b + |x(t) - X^b| \times A^+ \times L, i \leq \frac{pop_3}{2} \end{cases} \quad (42)$$

where  $pop_3$  denotes the number of small dung beetles,  $Q$  a random variable following a normal distribution,  $A$  is a  $1 \times D$  matrix, with matrix elements being either 1 or -1,  $A^+ = A^T(AA^T)^{-1}$ ,  $A^T$  is the transpose of  $A$ , and  $L$  is a  $1 \times D$  matrix with all elements equal to 1.

### 4) Adaptive t-Distribution Disturbance

For DBO, population diversity tends to decrease during the later iterations, making the algorithm prone to falling into local optima. Therefore, an adaptive t-distribution disturbance is applied to perturb individuals, enhancing their capability to escape local optima. Considering the randomness and computational complexity of mutation, only the global best individual is perturbed. The updated position after adaptive t-distribution mutation is:

$$x_i(t+1) = x_i(t) + x_i(t) \oplus \text{trnd}(t) \quad (43)$$

where  $\text{trnd}(\cdot)$  represents the adaptive t-distribution function, and  $t$  is the current iteration number.

## B. AEKF Fusion Positioning Method

EKF is a Kalman Filtering (KF) algorithm based on nonlinear systems. It linearizes the system by approximating nonlinear components and updates both the observation and state transition equations. In multi-sensor fusion, EKF can integrate the outputs of multiple sensors by leveraging their individual measurement models to estimate the system state, thereby achieving more accurate and stable localization. At present, EKF has already demonstrated promising results in Wi-Fi/PDR fusion-based indoor positioning. However,

traditional EKF assumes that the process noise covariance matrix ( $Q$ ) and the measurement noise covariance matrix ( $R$ ) are fixed values, which is not suitable in dynamic environments and negatively impacts positioning accuracy.  $Q$  represents the uncertainty in the system model (e.g., step length errors or disturbances), while  $R$  reflects the uncertainty in sensor measurements. If  $Q$  is too large, the filter will rely too heavily on measurements, possibly leading to instability; if  $Q$  is too small, the filter may respond too slowly. Similarly, a large  $R$  indicates high measurement uncertainty; a small  $R$  suggests a strong correlation in the measurements. Moreover, in Wi-Fi/PDR fusion positioning, the weight of the observation information is also critical to the system. Wi-Fi positioning is generally considered absolute, so in the initial stage, its observation weight is relatively large. However, in real-world environments, Wi-Fi signals are often affected by factors such as phone shaking and multipath effects, leading to significant measurement errors. Directly incorporating such inaccurate Wi-Fi observations into the fusion model may seriously degrade the overall positioning accuracy.

To address the limitations of traditional EKF in terms of adaptability and accuracy for indoor positioning, this paper proposes an AEKF method. This method introduces ADBO to dynamically adjust the  $Q$  and  $R$  matrices in the EKF, enabling the filter to adapt to system and environmental dynamics. At the same time, an adaptive weight adjustment method is introduced. When the system detects significant errors in Wi-Fi positioning results, the observation weight of Wi-Fi is reduced in the fusion model to minimize its negative impact on the final positioning outcome, thereby ensuring the stability and accuracy of the fusion-based localization. The implementation of AEKF involves four key steps: state prediction, covariance prediction, measurement update, and adaptive covariance adjustment using ADBO, and adaptive weight adjustment. The steps are as follows:

Step 1: State Prediction: Use the PDR algorithm to predict the current state and calculate the predicted value based on step length and heading. This step uses PDR data at peak step positions to provide an initial state estimate for the next Kalman filter iteration. The state transition function is used to predict the next state of the system. The current state equation is as follows:

$$X_{k|k-1} = f(X_{k-1|k-1}) + W_{k-1} = AX_{k-1|k-1} + W_{k-1} = \begin{bmatrix} x_{k-1} + l_{k-1} \sin \theta_{k-1} \\ y_{k-1} + l_{k-1} \cos \theta_{k-1} \\ \theta_{k-1} + \Delta \theta \end{bmatrix} + W_{k-1} \quad (44)$$

where  $X_{k|k-1}$  is the predicted estimate of the  $k$ -th state,  $f$  is the state transition function,  $x_k$  and  $y_k$  denote the position coordinates of the  $k$ -th state,  $\theta_k$  represents the heading angle of the  $k$ -th state,  $\Delta \theta$  is the change in heading angle, and  $W_{k-1}$  is the process noise.

After linearization, the Jacobian matrix of the state transition function is obtained as:

$$A = \frac{\partial f(X_{k-1|k-1})}{\partial X_{k-1|k-1}} = \begin{bmatrix} 1 & 0 & d_{k-1} \cos \theta_{k-1} \\ 0 & 1 & -d_{k-1} \sin \theta_{k-1} \\ 0 & 0 & 1 \end{bmatrix} \quad (45)$$

Step 2: Covariance Prediction: Calculate the state covariance matrix based on the predicted state value to measure the uncertainty in the prediction result. This

covariance matrix incorporates the errors from the PDR step length information and provides an estimate of the confidence in the predicted state. The predicted covariance matrix is updated as follows:

$$P_{k|k-1} = A \cdot P_{k-1|k-1} \cdot A^T + Q \quad (46)$$

where  $P_{k|k-1}$  is the predicted estimate of the covariance matrix for the  $k$ -th state, and its initial value of  $P_{k|k-1}$  is:

$$P_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (47)$$

$Q$  is the process noise covariance matrix, which is expressed as follows:

$$Q = \begin{bmatrix} \sigma_{px}^2 & 0 & 0 \\ 0 & \sigma_{py}^2 & 0 \\ 0 & 0 & \sigma_{yaw}^2 \end{bmatrix} \quad (48)$$

where  $\sigma_{px}^2$  and  $\sigma_{py}^2$  represent the standard deviations of the PDR process for the x-axis and y-axis positions, respectively, and  $\sigma_{yaw}^2$  represents the standard deviation of the heading angle.

Step 3: Measurement Update: Update the measurement using Wi-Fi fingerprint data. The measurement equation is:

$$Z_k = h(X_{k|k-1}) + V_k = \begin{bmatrix} x_W^k \\ y_W^k \\ \theta_{k-1} \\ l_{k-1} \end{bmatrix} + V_k \quad (49)$$

where  $h(\cdot)$  represents the observation function,  $x_W^k$  and  $y_W^k$  are the position coordinates of the  $k$ -th state obtained from Wi-Fi fingerprint positioning,  $V_k$  is the Gaussian noise in the measurement process.  $V_k$  and  $W_{k-1}$  are uncorrelated.

Calculate the Jacobian matrix of the observation function:

$$H = \frac{\partial h(X_{k|k-1})}{\partial X_{k-1|k-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (50)$$

The Kalman gain is given by:

$$K_K = P_{k|k-1} \cdot H^T \cdot (H \cdot P_{k|k-1} \cdot H^T + R)^{-1} \quad (51)$$

where  $R$  is the measurement noise covariance matrix, expressed as:

$$R = \begin{bmatrix} \sigma_{wx}^2 & 0 & 0 & 0 \\ 0 & \sigma_{wy}^2 & 0 & 0 \\ 0 & 0 & \sigma_l^2 & 0 \\ 0 & 0 & 0 & \sigma_{yaw}^2 \end{bmatrix} \quad (52)$$

where  $\sigma_{wx}^2$  and  $\sigma_{wy}^2$  are the standard deviations of the x-axis and y-axis in Wi-Fi fingerprint positioning, respectively,  $\sigma_l^2$  and  $\sigma_{yaw}^2$  are the standard deviations of the PDR step length and heading, respectively.

Update the state estimate using the observed measurements:

$$X_{k|k} = X_{k|k-1} + K_K \cdot (z_k - h(X_{k|k-1})) \quad (53)$$

Update the noise covariance matrix of the  $k$ -th state:

$$P_{k|k} = (I - K_K \cdot H) \cdot P_{k|k-1} \quad (54)$$

Step 4: Adaptive Covariance Adjustment Using ADBO: ADBO is used to dynamically adjust the  $Q$  and  $R$  matrices of EKF in real-time, minimizing the deviation between EKF estimates and actual values. Through multiple iterations, EKF achieves optimal  $Q$  and  $R$  matrices and thereby obtains the optimal state estimate. The core of ADBO is the fitness function, which constrains the optimization process to enable  $Q$  and  $R$  matrices to effectively reflect the dynamic changes in system and measurement noise. Therefore, the root mean square error (RMSE) is selected as the fitness function for the algorithm. By choosing RMSE as the fitness function, ADBO can find a more desirable solution in optimizing the EKF, thereby improving the accuracy and reliability of the model in indoor positioning. The specific definition is as follows:

$$f = \frac{1}{N} \sum_{k=1}^N (Z_k - h(X_{k|k-1}))^2 \quad (55)$$

where  $Z_k$  is the actual measurement value at time  $k$ ,  $h(X_{k|k-1})$  is the EKF's estimated observation model,  $N$  is the total number of observations. The  $Q$  and  $R$  matrices cannot be directly optimized by ADBO as a whole; each diagonal element in  $Q$  and  $R$  is optimized by ADBO. ADBO begins by randomly initializing the diagonal elements, and then iteratively updates them to the next generation, where each updated value is compared to obtain the optimal solution. The optimal values are then integrated to form the final optimized covariance matrices. This optimization process is repeated until the maximum number of iterations is reached. The adaptive covariance update steps of ADBO are as follows:

(1) Initialization: Initialize the standard deviations of the  $Q$  and  $R$  matrices as the diagonal elements, forming the initial dung beetle population. Each beetle is a candidate set of diagonal elements for  $Q$  and  $R$ , and these standard deviations are used as the initial values for the optimization process. Then, initialize fitness values, set the best fitness in the current generation as the current optimal position, and set the overall best fitness as the global optimum.

(2) Position Update: Update the positions of dung beetles based on the four behaviors defined in ADBO: rolling, breeding, foraging, and stealing. For example, the rolling behavior in ADBO is defined as:

$$Q_{11}(k) = \begin{cases} Q_{11}^b \cdot MOP \cdot ((Ub - Lb) \cdot \mu + Lb), \delta < m \\ |Q_{11}(k) + \tan \theta| |Q_{11}(k) - Q_{11}(k-1)|, \delta \geq m \end{cases} \quad (56)$$

(3) Matrix Generation: Construct the  $Q$  and  $R$  covariance matrices using the outputs of ADBO.

(4) State Update: Perform Use the optimized  $Q$  and  $R$  matrices with actual observation data to perform the EKF measurement update step and update the system state estimates.

(5) Fitness Calculation: Compute the current fitness value and update the current and global optimal positions.

(6) Iteration check: If the maximum number of iterations is reached, the algorithm terminates and outputs the optimal  $Q$  and  $R$  matrices; otherwise, return to (2).

Step 5: Adaptive Weight Adjustment: Before fusing Wi-Fi and PDR positioning results, the error between the Wi-Fi positioning and the PDR positioning result is compared with a threshold. The error function is defined as:

$$e(X) = Z - HX \quad (57)$$

where  $Z$  represents the real-time observation information, and  $HX$  represents the predicted state information.

Under normal circumstances, when  $e(X)$  is smaller than the set positioning difference threshold of  $\delta$ , it is assumed that the observation error meets the fusion requirements, and the observation weight in the fusion model remains unchanged. If  $e(X)$  exceeds  $\delta$ , the observation error is considered large. Given that PDR has strong autonomy and high short-term positioning accuracy, a method is needed to reduce the observation information weight, making the positioning result lean more towards PDR. To achieve this, an adaptive weight factor  $\mu$  is computed piecewise to dynamically adjust the weight of the observation information. Let the thresholds be set as  $k_0 = \delta$  and  $k_0 = 3\delta$ , then the values of  $\mu$  is given as follows:

$$\mu = \begin{cases} \frac{k_1 - e(X)}{k_1 - k_0}, & k_0 < e(X) < k_1 \\ 1, & e(X) < k_0 \\ \frac{1}{4}, & e(X) > k_1 \end{cases} \quad (58)$$

If the dynamic adjustment of the initial observation weight is  $R$ , then after applying the weight adaptation factor, the weight becomes  $\mu * R$ . In this paper, the position error threshold is set to 1.5 m, which corresponds to the walking distance between two Wi-Fi positions for a pedestrian. This ensures the effectiveness of the adaptive adjustment mechanism and prevents the positioning difference threshold from being set too small, which would fail to correct the accumulated PDR errors, thereby impacting the positioning accuracy of the fusion model. The workflow of AEKF is shown in Fig. 4.

### C. Positioning Evaluation Metrics

To comprehensively evaluate the performance of each positioning algorithm, select four commonly used error statistical indicators as evaluation metrics for system positioning accuracy: Average Positioning Error (APE), Standard Deviation (STD), RMSE, and Cumulative Distribution Function (CDF).

At the  $i$ -th point, the positioning error is measured by the Euclidean distance between the actual Position and its estimated counterpart, and is computed using the following formula:

$$E_i = \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (59)$$

APE: Reflects the average level of overall positioning deviation. The smaller the APE, the higher the positioning accuracy. The formula is:

$$MPE = \frac{1}{N} \sum_{i=1}^N E_i \quad (60)$$

STD: Measures the variability of positioning errors. A smaller STD indicates more stable results. The formula is as follows:

$$STD = \sqrt{\frac{1}{N} \sum_{i=1}^N (E_i - MPE)^2} \quad (61)$$

RMSE: Reflects the overall magnitude of error and is more sensitive to large errors. It is calculated using the following formula:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N E_i^2} \quad (62)$$

CDF: CDF is used to demonstrate the algorithm's performance under different error levels. The  $x$ -axis represents the positioning error, while the  $y$ -axis represents the cumulative probability. The slope of the curve reflects positioning accuracy: the steeper the slope, the higher the cumulative probability within that error range, and the better positioning accuracy. The applied formula is shown below:

$$CDF(c) = p(E \leq c) \quad (63)$$

where  $CDF(c)$  denotes the probability that the positioning error  $E \leq c$  occurs, and  $c$  is a given constant.

## V. EXPERIMENT AND RESULT ANALYSIS

### A. Experimental Environment

In this experiment, eight MERCURY-MW305R routers were used as Wi-Fi signal transmitters, and a Huawei smartphone equipped with RSSI collection software and PDR data acquisition tools was used as the receiver. The test area consisted of an L-shaped corridor with a length of 15m in the north-south direction and a width of 15m in the east-west direction. The deployment scenario is shown in Fig. 5. Eight APs were installed as signal sources at a height of approximately 2.5m above the ground. After setting up the

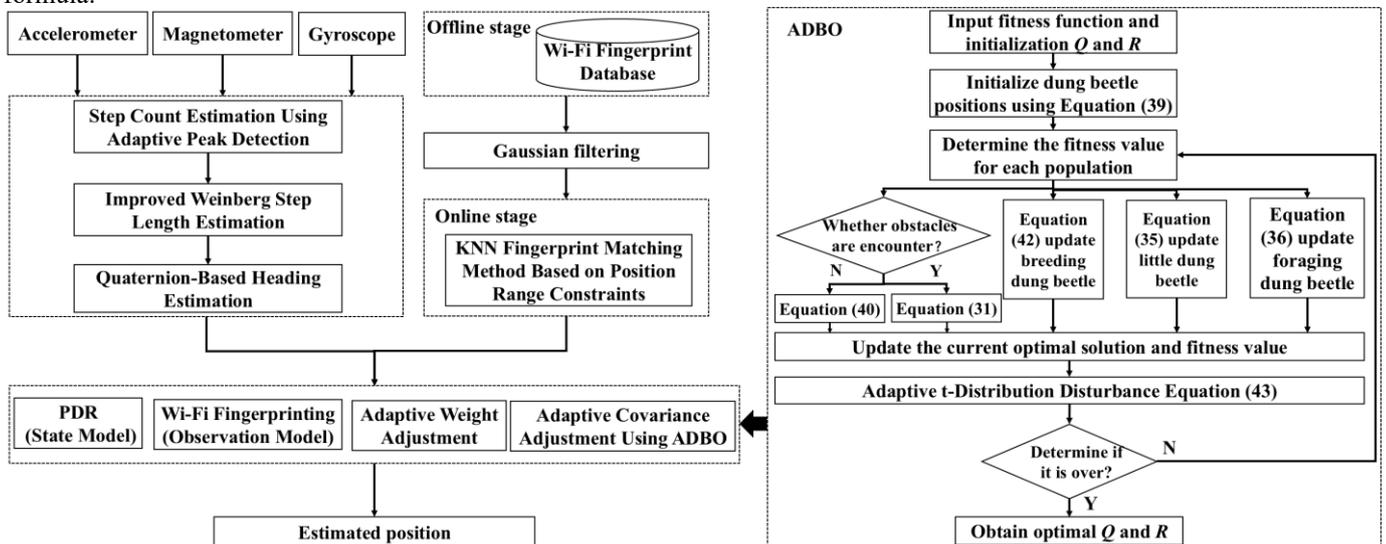


Fig. 4. Flowchart of the AEKF-based fusion positioning algorithm.

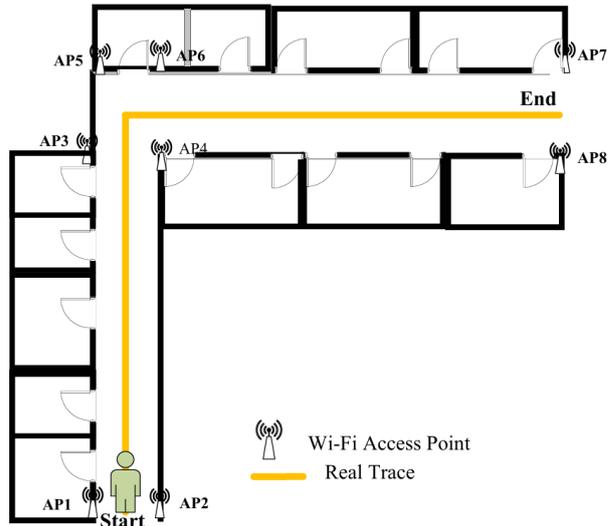


Fig. 5. Experimental Scenario.

signal sources, RPs were established using a grid-based method. The corridor area was uniformly divided into grids with a spacing of  $1\text{ m} \times 1\text{ m}$ , and the intersections of the grid lines formed the RPs. A total of 104 RPs were deployed. Offline RSSI data were collected at each RP to construct the Wi-Fi fingerprint database by mapping RSSI values to corresponding coordinates. During the online phase, the experimenter held the smartphone flat against the chest and walked at a constant speed to facilitate consistent data collection across position states. Both Wi-Fi and PDR data were collected at a sampling rate of 50 Hz. The total number of steps taken was 30, resulting in 31 position states.

*B. Performance Evaluation of ADBO*

To thoroughly evaluate the optimization performance and robustness of ADBO, it is compared with four other algorithms: the multi-strategy improved Dung Beetle Optimization algorithm (CRCDBO) [28], chaotic map DBO (CMDBO) [29], DBO [26], SSA [30] and Black Kite Algorithm (BKA) [31]. These six algorithms are tested on the

first 13 benchmark functions of the CEC2005 suite [32]. Among these functions, F1-F7 are unimodal test functions used to evaluate the convergence speed and accuracy of the algorithms, while F8-F13 are multimodal functions with a single global optimum and multiple local optima, designed to evaluate the algorithm’s ability to escape local optima. The names, dimensions, search ranges, and optimal values of the CEC2005 benchmark functions are reference in [32]. To ensure experimental reliability, each algorithm was independently run 30 times, and the average convergence accuracy and standard deviation were used as evaluation metrics. The experimental results are shown in TABLE I, where bold values indicate the best performance.

From the results in TABLE I, ADBO is observed to A outperforms the five comparison algorithms in terms of optimization accuracy and achieves optimal results for all 13 CEC2005 benchmark functions. When solving functions F1, F2, F3, F4, F9, F10 and F11, both ADBO and CRCDBO reached theoretical optimal values for mean convergence

TABLE I  
COMPARISON OF RESULTS FOR DIFFERENT ALGORITHMS

NO	Criterion	ADBO	CRCDBO	CMDBO	DBO	SSA	BKA
F1	Mean	<b>0</b>	<b>0</b>	7.40E-142	8.11E-126	1.00E-66	8.56E-100
	Std	<b>0</b>	<b>0</b>	1.04E-141	1.14E-125	1.42E-66	1.21E-99
F2	Mean	<b>0</b>	<b>0</b>	4.66E-68	1.42E-66	1.12E-31	4.27E-47
	Std	<b>0</b>	<b>0</b>	8.08E-68	2.46E-66	1.93E-31	8.16E-47
F3	Mean	<b>0</b>	<b>0</b>	4.07E-122	6.91E-117	1.63E-30	1.09E-97
	Std	<b>0</b>	<b>0</b>	7.02E-122	1.19E-116	2.83E-30	1.55E-97
F4	Mean	<b>0</b>	<b>0</b>	1.50E-54	2.37E-54	1.73E-36	1.38E-52
	Std	<b>0</b>	<b>0</b>	2.60E-54	2.60E-54	2.81E-36	2.39E-45
F5	Mean	<b>4.78E-07</b>	4.30E-04	2.57E+01	2.56E+01	1.03E-06	2.78E+01
	Std	<b>7.89E-07</b>	7.32E-04	1.71E-01	5.28E-01	8.25E-07	8.96E-01
F6	Mean	<b>1.23E-14</b>	1.61E-11	1.04E-03	1.24E-03	2.08E-11	1.07E+00
	Std	<b>1.09E-14</b>	1.48E-11	3.13E-04	6.06E-04	1.80E-11	2.09E-01
F7	Mean	<b>8.49E-05</b>	1.44E-04	1.02E-03	1.37E-03	1.72E-03	3.78E-04
	Std	<b>5.00E-05</b>	1.62E-04	8.80E-04	9.45E-04	2.69E-03	1.79E-04
F8	Mean	<b>1.26E+04</b>	1.14E+04	1.11E+04	8.23E+03	8.28E+03	9.76E+03
	Std	<b>2.81E-10</b>	1.87E+03	2.32E+03	2.32E+02	7.47E+02	1.20E+03
F9	Mean	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
F10	Mean	<b>4.44E-16</b>	<b>4.44E-16</b>	<b>4.44E-16</b>	<b>4.44E-16</b>	<b>4.44E-16</b>	<b>4.44E-16</b>
	Std	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
F11	Mean	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
F12	Mean	<b>2.03E-14</b>	4.24E-12	1.22E-04	5.19E-05	1.09E-12	8.46E-02
	Std	<b>2.37E-14</b>	6.84E-12	4.02E-05	4.36E-05	1.77E-12	4.02E-02
F13	Mean	<b>4.88E-14</b>	5.23E-11	7.84E-03	1.40E-02	5.60E-12	1.81E+00
	Std	<b>8.20E-14</b>	7.53E-11	6.25E-03	1.05E-02	2.80E-12	5.11E-01

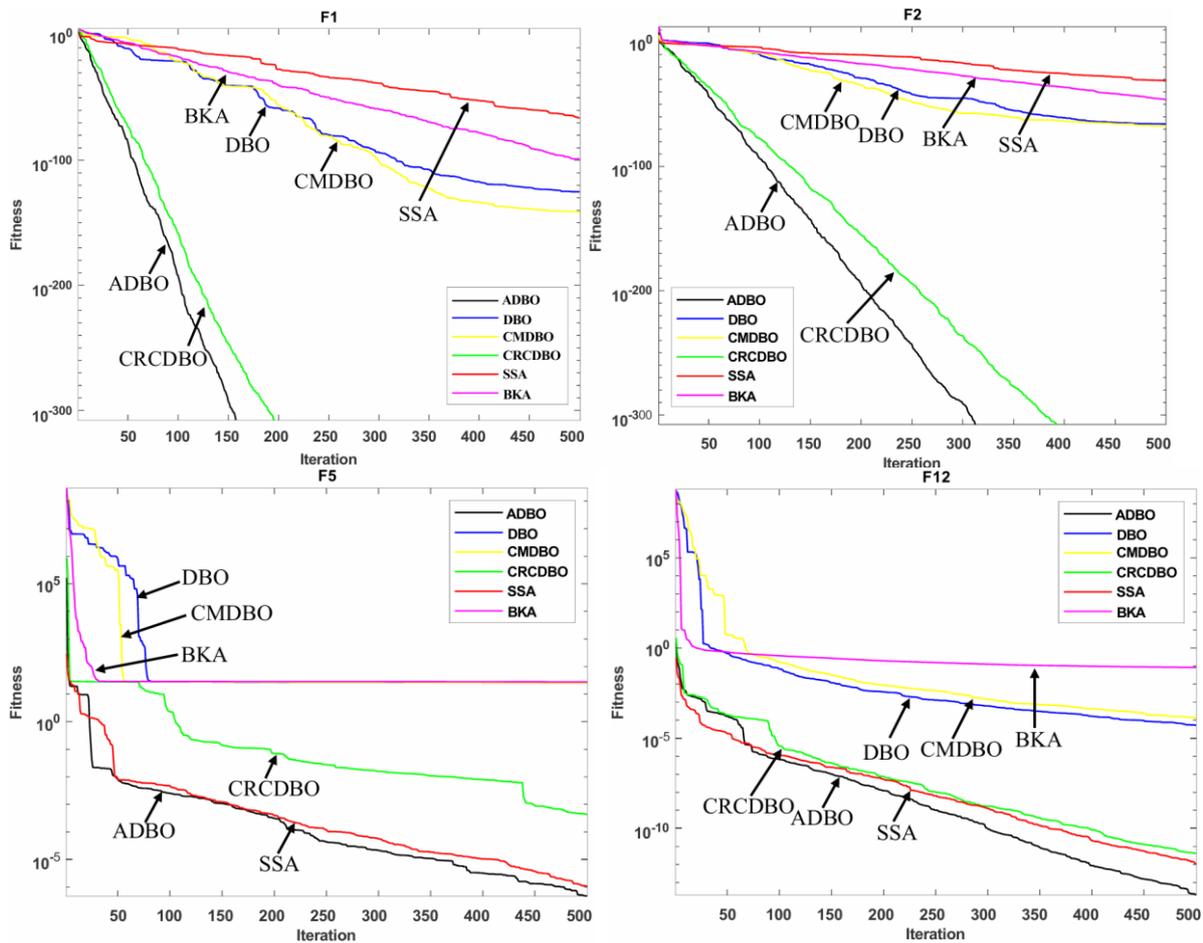


Fig. 6. Convergence curves of different algorithms on selected benchmark functions.

accuracy and standard deviation, significantly surpassing the other algorithms. For functions F5、F6、F7、F12 and F13, ADBO achieved improvements of 1 to 14 orders of magnitude compared to the other five algorithms, with both its average convergence accuracy and standard deviation ranking first, better than all the others. For the F8 function, only ADBO achieved the theoretical optimal value. To visually demonstrate the difference in optimization capabilities between ADBO and the other five intelligent algorithms, the convergence curves for some benchmark functions are shown in Fig. 6. From Fig. 6, it is evident that whether it is a unimodal or multimodal test function, ADBO's convergence curve is consistently at the lowest position, and compared to other algorithms, ADBO achieves good optimization accuracy with fewer iterations. Therefore, ADBO's global search capability and ability to escape local optima are generally superior to those of the other algorithms. Based on the experimental results from TABLE I and Fig.6, it can be concluded that ADBO exhibits excellent optimization performance and stability in both global exploration and local exploitation, with superior optimization accuracy, solution speed, and robustness when solving different test functions compared to DBO, other improved DBO algorithms, and other algorithms, validating ADBO's effectiveness in solving complex problems. Since tuning of the matrices  $Q$  and  $R$  in EKF is essentially a parameter optimization problem, and swarm intelligence optimization algorithms are an efficient means for solving such problems. Therefore, ADBO was used to optimize  $Q$  and  $R$  in EKF in this study.

C. Analysis of Position Range-Based Restricted KNN Fingerprint Matching Method

1) Analysis of K Value Selection

In the KNN fingerprint matching method based on position range constraints, the selection of parameter K has a significant impact on positioning accuracy [33]. A smaller K may be affected by RSSI noise and outliers, resulting in larger positioning errors. Conversely, a larger K may introduce reference points that are far from the target location, reducing positioning accuracy. Therefore, an experimental analysis is conducted to evaluate the effect of different K values on positioning accuracy to determine the optimal range for K. The selection of K is shown in Fig. 7.

In the process of analyzing the selection of K value, it was found through multiple sets of experimental data that the optimal positioning results can be obtained when K is between 3 and 5. Therefore, in the subsequent experiments of this paper, the value of K is set to 5.

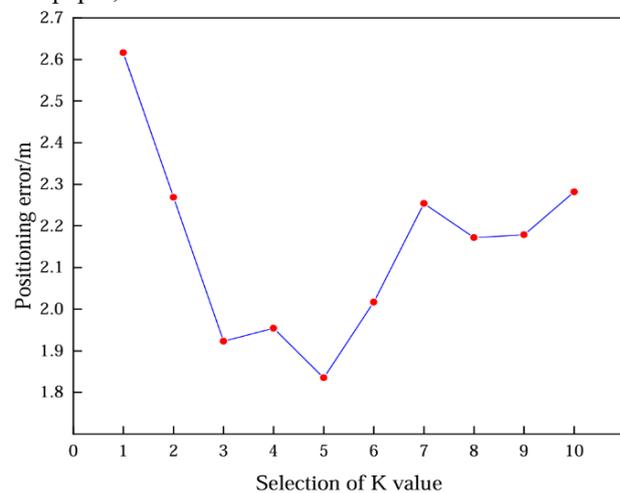


Fig. 7. Selection of K Value.

2) *KNN Fingerprint Matching Method Based on Position Range Constraints*

To verify the positioning performance of the KNN fingerprint matching method based on position range constraints, the positioning results were compared with KNN [34], Random Forest (RF) [35], and MLP [14]. The specific results are shown in Fig. 8, and TABLE II summarizes the comparison of APE, RMSE, and positioning time for each algorithm. The experimental results show that the KNN fingerprint matching method based on position range constraints effectively reduces interference caused by noise and position error during the fingerprint matching process by introducing spatial constraints based on known positions, thereby improving positioning accuracy. Moreover, it maintains a low level of computational complexity, demonstrating strong real-time advantages. Therefore, the KNN fingerprint matching method based on position range constraints achieves a good balance between accuracy and real-time performance.

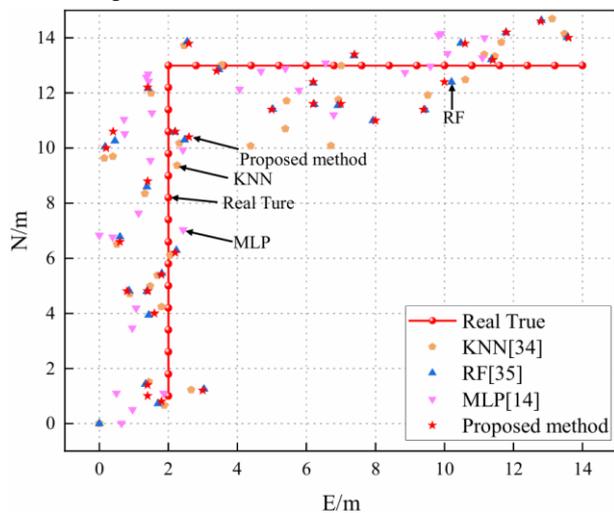


Fig. 8. Results of Four Wi-Fi Fingerprint Positioning Algorithms.

TABLE II  
PERFORMANCE COMPARISON OF WI-FI FINGERPRINT MATCHING ALGORITHMS

Method	APE/m	RMSE/m	Positioning
			Time/s
KNN	2.033	2.874	28.6
RF	1.879	2.693	41.7
MLP	1.845	2.623	52.4
Proposed method	1.835	2.603	34.8

1) *Step Count Estimation Using Adaptive Peak Detection*

In this experiment, the adaptive peak detection method is used for step estimation to ensure capture of step counts. As shown in Fig. 9, the adaptive peak detection method significantly improves step detection accuracy compared to the traditional method. The adaptive approach achieved nearly 100% accuracy in step counting, while the traditional method detects 5 false steps beyond the actual number of steps taken. In contrast, the adaptive method effectively reduces false step detections caused by external interference, thereby enhancing both the accuracy and robustness of PDR positioning.

2) *Improved Weinberg Step Length Estimation*

An improved Weinberg step length estimation method is adopted, which estimates step length by calculating the difference between the maximum and minimum acceleration values, and introduces acceleration variance as an additional feature to enhance the model's adaptability and robustness. To verify the accuracy of the model, a subject was instructed to walk at a constant speed with a fixed step length of 0.6 m. Fig. 10 shows the step length estimation results obtained from four different models: Weinberg, Scarlet, Kim, and the improved Weinberg model. The improved Weinberg demonstrates superior accuracy and stability. Compared with the other three models, it yields smaller errors and exhibits smoother curve fluctuations. Based on experimental findings, the values of K and B are assigned as 0.364 and 0.105, respectively.

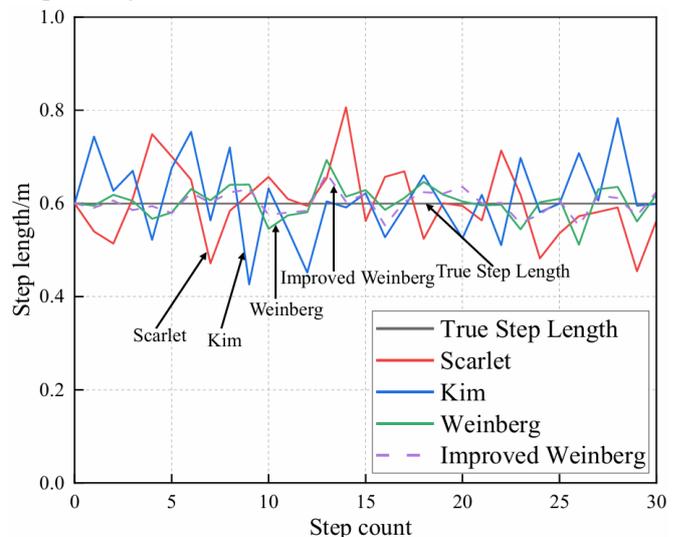


Fig. 10. Step Length Estimation Results.

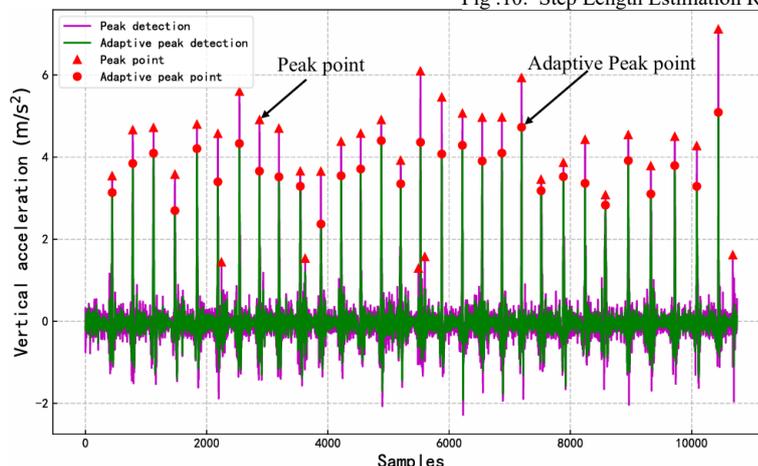


Fig. 9. Results of Four Wi-Fi Fingerprint Positioning Algorithms.

E. Fusion Positioning Experiment Analysis

Wi-Fi fingerprint positioning is a single-point positioning method and does not suffer from cumulative errors. However, due to the instability of Wi-Fi signal reception and the low sampling frequency, Wi-Fi fingerprint positioning results often show position jumps, which are reflected in the trajectory as discontinuous and unsmooth paths. In contrast, PDR positioning is a relative positioning method. It offers high sampling frequency and trajectory continuity, resulting in smoother positioning results. However, it is prone to cumulative error over time. To address these limitations, AEKF is used to fuse the positioning results from Wi-Fi fingerprinting and PDR to achieve more accurate pedestrian localization. The fusion experiment setup is consistent with

the previous Wi-Fi fingerprinting and PDR experiments. In an L-shaped corridor, the subject walks at a constant speed while collecting Wi-Fi RSSI and smartphone-based PDR sensor data for positioning.

The experiment compares the performance of Wi-Fi fingerprint positioning, PDR, KF, EKF [8], the method in reference [14], AEKF optimized by DBO (DEKF), and this paper proposed AEKF fusion positioning algorithm. The positioning trajectories in the L-shaped corridor are shown in Fig. 11, and the single-point positioning errors are presented in Fig. 12. TABLE III presents a comparison of APE, STD, RMSE, maximum single-point error, and minimum single-point error under different algorithms under the same experimental environment.

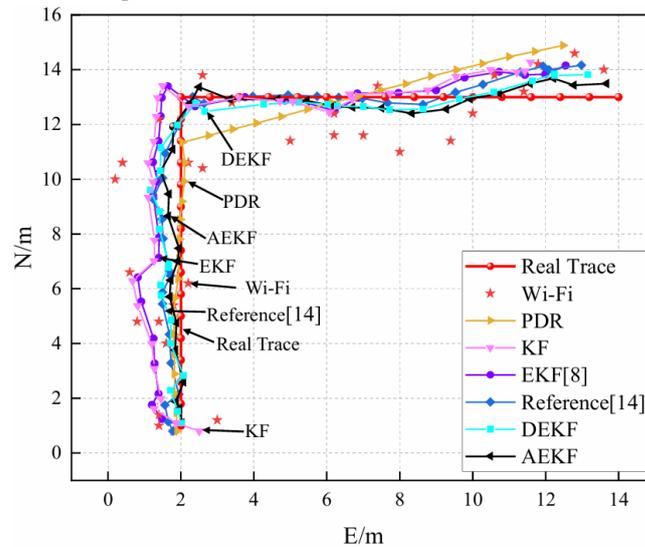


Fig. 11. Comparison of Positioning Trajectories.

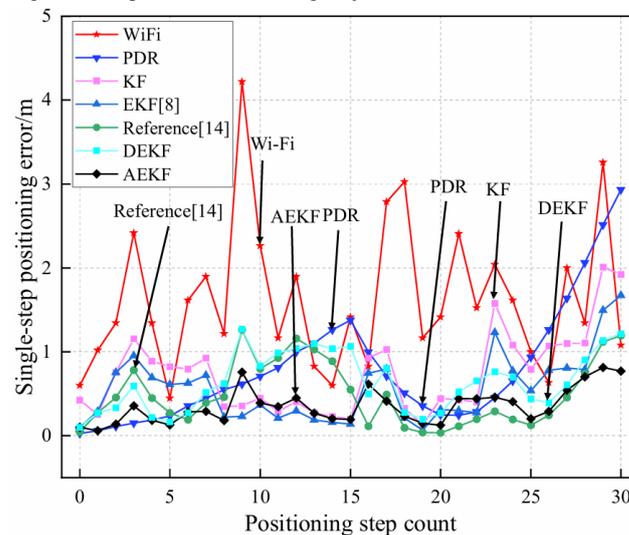


Fig. 12. Single Point Positioning Error.

TABLE III  
AVERAGE POSITIONING ERROR OF DIFFERENT ALGORITHMS

METHOD	APE/m	STD/m	RMSE/m	MAXIMUM SINGLE-POINT ERROR/m	MINIMUM SINGLE-POINT ERROR/m
Wi-Fi	1.835	2.652	2.603	4.219	0.447
PDR	1.264	1.861	2.250	2.933	0.025
KF	1.163	1.677	2.041	2.009	0.129
EKF [8]	1.109	1.478	1.848	1.672	0.068
Reference [14]	0.966	1.275	1.600	1.193	0.036
DEKF	0.751	1.096	1.329	1.269	0.101
AEKF	0.627	0.968	0.814	0.814	0.060

As shown in Fig. 11 and Fig. 12 and TABLE III, Wi-Fi positioning results exhibit significant error fluctuations and severe position jumps. The APE is 1.837 m. The PDR positioning suffers from cumulative errors as steps increase, reaching a final error of 2.933 m and an APE of 1.264 m. In comparison, KF and EKF alleviate some of the errors by fusing Wi-Fi and PDR data, reducing the APE to 1.163 m and 1.109 m, respectively. The method in reference [14], which dynamically adjusts the weights of individual sources within the EKF framework, achieves more effective multi-source fusion with an APE of 0.966 m. Although this method improves accuracy, the error fluctuations are still not fully suppressed. To further enhance the accuracy and robustness of fusion positioning, this paper incorporates intelligent optimization into the EKF framework. Specifically, two optimization variants using DBO and ADBO are proposed to dynamically tune the EKF state covariance matrix, improving the filter's adaptability to environmental changes. Additionally, an adaptive weighting mechanism dynamically adjusts the fusion ratio between Wi-Fi and PDR based on real-time observation quality, further enhancing system stability in varying conditions. Among them, the DEKF improves the error propagation process to a certain extent, achieving an APE of 0.751 m, and outperforms other fusion methods across multiple metrics. In contrast, the AEKF optimized by ADBO retains strong global search capabilities and further enhances the convergence precision of the covariance matrix through strategies such as good point set initialization and adaptive  $t$ -distribution perturbation, demonstrating superior robustness and adaptability to dynamic environments. Experimental results demonstrate that AEKF achieves APE of 0.627 m, STD of 0.814 m, RMSE of 1.027 m, a maximum error of 1.193 m, and a minimum error of 0.060 m. It effectively suppresses both Wi-Fi position jumps and PDR cumulative drift, significantly improving overall positioning accuracy and stability.

The APE only reflects the mean level of accuracy and does not capture the overall error distribution, making it insufficient to fully evaluate the robustness of the positioning system. To provide a more comprehensive evaluation of the system's robustness, the CDF is introduced for analysis. Fig. 13 shows a comparison of the CDF curves for AEKF and six other methods. In terms of high-precision performance within 1 m, AEKF has 80% of the errors within 1 m. DEKF has 76% of the errors within 1 m, the method in reference [14] has 71% within 1 m, KF has 65% within 1 m, EKF and PDR each have 64% within 1 m, while Wi-Fi has only 40% of the errors within 1 m. These results demonstrate that the proposed method outperforms all six alternatives in both accuracy and robustness. In addition, AEKF achieves the lowest maximum positioning error of 1.21 m. The maximum errors of the method in reference [14], DEKF, EKF, KF, PDR, and Wi-Fi are 1.36 m, 1.51 m, 1.73 m, 1.98 m, 2.33 m, and 2.35 m, respectively. These findings indicate that AEKF exhibits a more concentrated CDF curve across different positions, along with faster convergence and greater stability, confirming its superior error distribution performance compared to the other methods.

Overall, AEKF exhibits strong performance in various respects. Compared with other algorithms, AEKF leverages adaptive ADBO-based adaptive covariance adjustment and

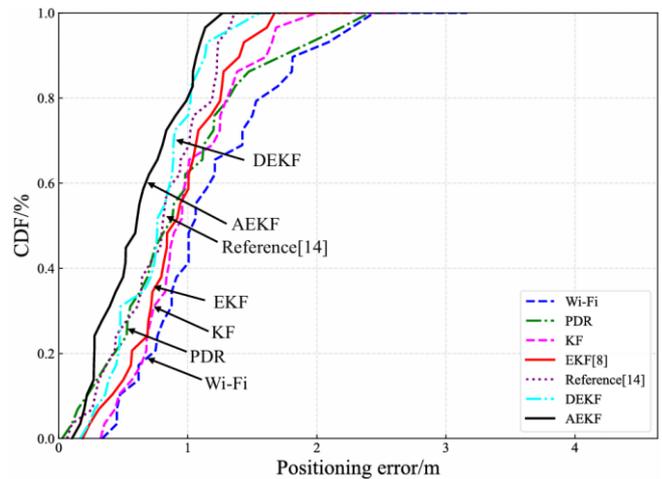


Fig. 13. CDF of Positioning Errors.

adaptive weight adjustment mechanisms, enabling better adaptability to complex and dynamic indoor environments, while effectively integrates multi-source positioning information from Wi-Fi and PDR. AEKF achieves superior results across multiple error evaluation metrics and CDF distribution curves, demonstrating its promising potential for improving both accuracy and stability in indoor positioning applications.

## VI. CONCLUSION

To address the challenges of large positioning error fluctuations in Wi-Fi fingerprint positioning and the cumulative errors over time in PDR, this paper proposes an AEKF based indoor positioning algorithm integrating Wi-Fi and PDR. In the Wi-Fi positioning component, a KNN fingerprint matching method based on position range constraints is introduced to enhance the continuity of positioning points and effectively suppress error jumps in Wi-Fi positioning. In the PDR positioning component, an adaptive peak detection, an improved Weinberg step length estimation method, and quaternion-based heading calculation are employed. These enhancements not only improve step detection accuracy and adaptivity in step length estimation but also avoid the gimbal lock problem inherent in Euler angle calculations, thereby reducing cumulative errors. During the fusion positioning, an ADBO-based dynamic optimization is introduced to adaptively adjust the state covariance matrix in EKF, enhancing the filter's adaptability to environmental changes. At the same time, an adaptive weight adjustment mechanism is incorporated to dynamically tune the observation weights based on the quality of Wi-Fi and PDR measurements, further enhancing the robustness and stability of multi-source information fusion. Experimental results demonstrate that the proposed AEKF fusion positioning method significantly improves positioning accuracy and stability in typical indoor environments.

Future research will aim to incorporate more types of multimodal sensor data, such as UWB, Bluetooth, and barometers, to leverage the complementary strengths of various sensors in different environments. Additionally, the multi-source information weighting strategy will be further optimized to enhance positioning accuracy and adaptability in complex and dynamic scenarios. These improvements are expected to contribute to more reliable and scalable solutions for real-world applications.

## REFERENCES

- [1] Xuyin Wang, Yi Yao, Jiale Lin, Dong Chen, Ying Cai, Siquan Li, Yanling Shang, and Fangzheng Gao, "Indoor Localization Method Enhanced by an Improved Harris's Hawk Optimization-Based Particle Filter," *IAENG International Journal of Computer Science*, vol. 51, no. 3, pp. 267-275, 2024.
- [2] Zhang L, Zheng X, Chen Y, Lu H and Zhang C, "Indoor fingerprint localization algorithm based on WKNN and LightGBM-GA," *Measurement Science and Technology*, vol. 35, no. 11, pp. 116313, 2024.
- [3] Chunhua Zhu, Manyi Li, and Mingyang Cui, "Application of A Long Short-Term Memory Neural Network Algorithm Fused with ResNet in UWB Indoor Positioning," *IAENG International Journal of Computer Science*, vol. 52, no. 5, pp. 1417-1423, 2025.
- [4] Ji M, Ren G, Zhang H, and Ren R, "Collaborative positioning for emergency rescuers based on INS, GPS and ZigBee," *Physica Scripta*, vol. 99, no. 6, pp. 065530, 2024.
- [5] Wan Q, Wu T, Zhang K, Liu X, Cheng K, Liu J, and Zhu J, "A high precision indoor positioning system of BLE AOA based on ISSS algorithm," *Measurement*, vol. 224, pp. 113801, 2024.
- [6] Hou B and Wang Y, "RF-KELM indoor positioning algorithm based on WiFi RSS fingerprint," *Measurement Science and Technology*, vol. 35, no. 4, pp. 045004, 2024.
- [7] Li Z, Junna S, and Liao W, "A robust factor graph framework for navigation on PDR/magnetic field integration," *Measurement*, vol. 245, pp. 116509, 2025.
- [8] Zhu M, Jin M, Liu J, and Wang Z, "Novel MEMS-IMU/Wi-Fi integrated indoor pedestrian location algorithm," *Engineering Letters*, vol. 31, no. 2, pp. 774-781, 2023.
- [9] Qian Y and Chen X, "An improved particle filter based indoor tracking system via joint Wi-Fi/PDR localization," *Measurement Science and Technology*, vol. 32, no. 1, pp. 014004, 2020.
- [10] Mehrabian H and Ravanmehr R, "Sensor fusion for indoor positioning system through improved RSSI and PDR methods," *Future Generation Computer Systems*, vol. 138, pp. 254-269, 2023.
- [11] Liu Z, Song S, Chen J, and Hou C, "Enhanced WiFi/Pedestrian dead reckoning indoor localization using Artemisinin optimization-particle swarm optimization-particle filter," *Electronics*, vol. 13, no. 17, pp. 3366, 2024.
- [12] Chen J, Song S, and Liu Z, "A PDR/WiFi indoor navigation algorithm using the federated particle filter," *Electronics*, vol. 11, no. 20, pp. 3387, 2022.
- [13] D. Yu, C. Li and J. Xiao, "Neural Networks-Based Wi-Fi/PDR Indoor Navigation Fusion Methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-14, 2023.
- [14] Xu H, Meng F, Liu H, Shao H, and Sun L, "An adaptive multi-source data fusion indoor positioning method based on collaborative Wi-Fi fingerprinting and PDR techniques," *IEEE Sensors Journal*, vol. 24, pp. 31481-31494, 2024.
- [15] Pan L, Zhang H, Zhang L, Gao R, and Zhang Q, "Indoor positioning fingerprint database construction based on CSA-DBSCAN and RCVAE-GAN," *Physica Scripta*, vol. 99, no. 5, pp. 055002, 2024.
- [16] Zhou P, Wang H, Gravina R, and Sun F, "WIO-EKF: Extended Kalman filtering-based Wi-Fi and inertial odometry fusion method for indoor localization," *IEEE Internet of Things Journal*, vol. 11, pp. 23592-23603, 2024.
- [17] Shao K, Li Z, Guo Q, Liu Z, and Wu Q, "Fingerprint accuracy indication geomagnetic/PDR indoor adaptive integrated positioning based on smartphones," *IEEE Sensors Journal*, vol. 23, no. 23, pp. 29545-29558, 2023.
- [18] Huang G, Liu J, Yang S, Gong X, and Zhao Y, "Context-assisted personalized pedestrian dead reckoning localization with a smartphone," *Geo-spatial Information Science*, pp. 1-17, 2024.
- [19] Yang Y, Huang B, Xu Z, and Yang R, "A fuzzy logic-based energy-adaptive localization scheme by fusing WiFi and PDR," *Wireless Communications and Mobile Computing*, no. 1, pp. 9052477, 2023.
- [20] Wu L, Guo S, Han L, and Baris C. A., "Indoor positioning method for pedestrian dead reckoning based on multi-source sensors," *Measurement*, vol. 229, pp. 114416, 2024.
- [21] Bai S, Wen W, Li Y, Shi C, and Hsu L. T., "Towards persistent spatial awareness: A review of pedestrian dead reckoning-centric indoor positioning with smartphones," *IEEE Transactions on Instrumentation and Measurement*, no. 73, pp. 1-23, 2024.
- [22] Han Y, Yu X, Zhu P, Xiao X, Wei M, and Xie S, "A fusion positioning method for indoor geomagnetic/light intensity/pedestrian dead reckoning based on dual-layer tent-atom search optimization-back propagation," *Sensors*, vol. 23, no. 18, pp. 7929, 2023.
- [23] Zhao G, Wang X, Zhao H, and Jiang Z, "An improved pedestrian dead reckoning algorithm based on smartphone built-in MEMS sensors," *AEU-International Journal of Electronics and Communications*, vol. 168, pp. 154674, 2023.
- [24] Saadatzaheh E, Ali Abbaspour R, and Chehreghan A, "An improvement in smartphone-based 3D indoor positioning using an effective map matching method," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 10, pp. 13741-13771, 2023.
- [25] Magsi A. H., Diez L. E., and Knauth S, "Continuous high-precision positioning in smartphones by FGO-based fusion of GNSS-PPK and PDR," *Micromachines*, vol. 15, no. 9, pp. 1141, 2024.
- [26] Ji X, Qu Q, Ren Y. L., Lian J. X., and Jiang T. R., "Improved DBO algorithm incorporating disorienting behavior and dynamic population strategy for engineering problem solving," *Engineering Letters*, vol. 33, no. 1, pp. 90-103, 2025.
- [27] Dhal K. G., Sasmal B., Das A., Ray S., and Rai R., "A comprehensive survey on arithmetic optimization algorithm," *Archives of Computational Methods in Engineering*, vol. 30, no. 5, pp. 3379-3404, 2023.
- [28] Song D, Feng G, Qi T, Wang H, Pan D, and Zhang L, "A new combined transient electromagnetic noise reduction method of VMD-SVD-WTD based on improved dung beetle optimization algorithm with multi-strategy fusion," *Measurement Science and Technology*, vol. 36, no. 1, pp. 016105, 2024.
- [29] Huang H, Yao Z, Wei X, and Zhou Y, "Twin support vector machines based on chaotic map dung beetle optimization algorithm," *Journal of Computational Design and Engineering*, vol. 11, no. 3, pp. 101-110, 2024.
- [30] Xue J, and Shen B. "A novel swarm intelligence optimization approach: sparrow search algorithm." *Systems science & control engineering*, vol. 8, no. 1, pp. 22-34, 2020
- [31] Wang J, Wang W. C., Hu X. X., Qiu L, and Zang H. F., "Black-winged kite algorithm: A nature-inspired meta-heuristic for solving benchmark functions and engineering problems," *Artificial Intelligence Review*, vol. 57, no. 4, pp. 98, 2024.
- [32] Purba Daru Kusuma, and Anggunmekha Luhur Prasasti, Best-Other Algorithm: A Metaheuristic Combining Best Member with Other Entities," *IAENG International Journal of Computer Science*, vol. 51, no. 6, pp. 626-633, 2024
- [33] Yang Y, Yang Q, Zhang T, and Huang W, "A Brownian motion restricted K-nearest neighbor algorithm for indoor positioning," *Wireless Personal Communications*, vol. 139, no. 1, pp. 625-651, 2024.
- [34] Zuhairi M. F., Jikaraji A., Dao H., and Sulong A., "Analysis of an indoor positioning system for forklifts using WiFi fingerprinting," in *Proceedings of the 2024 International Visualization, Informatics and Technology Conference (IVIT)*, pp. 134-137, 2024.
- [35] Gorjan H. E., and Jiménez V. P. G., "Improving indoor WiFi localization by using machine learning techniques," *Sensors*, vol. 24, no. 19, pp. 6293, 2024.

**Xin-Peng Zheng** was born in 2000 in Dengzhou, Henan, China. He obtained his bachelor's degree from Guizhou Institute of Technology in 2023. Currently, he is pursuing a master's degree at the Key Laboratory of Advanced Manufacturing and Automation Technology (Guilin University of Technology). His research interests include intelligent sensor information processing technology and indoor positioning.

**Lie-Ping Zhang** received the PhD degree from Kunming University of Science and Technology, Kunming, China, in 2011. He is currently a professor of School of Mechanical Engineering, Guilin University of Aerospace Technology, Guilin, China. His research interests include wireless sensor network and system optimization & scheduling.

**Zhen-Tao Yu** obtained his bachelor's degree from Nanchang Hangkong University Science and Technology College in 2021. He is currently pursuing a master's degree at the Key Laboratory of Advanced Manufacturing and Automation Technology (Guilin University of Technology). His research interests include intelligent sensor information processing technology and intelligent algorithms.

**Cui Zhang** received Master degree from Guilin University of Technology, Guilin, China, in 2012. She is currently an Associate Professor in School of Information Engineering, Worked in Nanning College of Technology. Her research interests include Artificial Intelligence and Intelligent Control.