EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems Research Article EAI.EU

# An Efficient Method for BLE Indoor Localization Using Signal Fingerprint

Trong-Thanh Han[1,*], Phuc Nguyen Dinh[1], Toan Nguyen Duc[1], Vu Nguyen Long[1] and Hung Tan Dinh [2]

[1]School of Electronical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi 100000, Vietnam

[2]School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi 100000, Vietnam

## Abstract

The rise of Bluetooth Low Energy (BLE) technology has opened new possibilities for indoor localization systems. However, extracting fingerprint features from the Received Signal Strength Indicator (RSSI) of BLE signals often encounters challenges due to significant errors and fluctuations. This research proposes an approach that integrates signal filtering and deep learning techniques to improve accuracy and stability. A Kalman filter is employed to smooth the RSSI values, while Autoencoder and Convolutional Autoencoder models are utilized to extract distinctive fingerprint features. The system compares random test points with a reference database using normalized cross-correlation. Performance is assessed based on metrics such as the number of reference points with the highest cross-correlation ($k$), average localization error, and other statistical indicators. Experimental results show that the combination of the Kalman filter with the Convolutional Autoencoder model achieves an average error of 0.98 meters with $k = 4$. These findings indicate that this approach effectively reduces signal noise and enhances localization accuracy in indoor environments.

**Keywords:** Indoor Localization, Fingerprint, Bluetooth Low Energy, Autoencoder.

Received on 10 07 2024, accepted on 13 11 2024, published on 22 11 2024



doi: 10.4108/eetinis.v12i1.6571

## 1. Introduction

Indoor localization has long been a crucial issue in various large-scale applications today, such as inventory management, equipment tracking, and product monitoring. Currently, there are numerous technologies for indoor positioning, including Wi-Fi, Ultra-Wideband, Bluetooth, optical technology, and infrared. In [1], the authors highlighted the advantages of using Wi-Fi, UWB, and other technologies in the context of indoor positioning systems. Wi-Fi leverages existing infrastructure, reducing investment costs while enabling enhancements in accuracy, particularly with recent developments like Wi-Fi 6, which can achieve centimeter-level precision. Additionally, Wi-Fi operates effectively in densely populated environments due to its higher frequency, which minimizes interference from human energy absorption. In contrast, UWB offers positioning accuracy below 10 cm and excels in handling multipath conditions, remaining largely unaffected by obstacles such as walls. The high data transmission rate and temporal resolution of UWB further improve its effectiveness in applications that demand high precision.

However, these technologies also present some drawbacks compared to Bluetooth Low Energy (BLE). Wi-Fi consumes significantly more power (216.71 $mW$ compared to BLE's 0.367 $mW$), potentially reducing the battery life of mobile devices. Additionally, Wi-Fi can suffer from interference in densely populated environments, leading to a decrease in positioning accuracy. On the other hand, UWB typically requires higher infrastructure investment due to the need for multiple nodes, and its high energy consumption

* Corresponding author. Email: thanh.hantrong@set.hust.edu.vn

limits the viability of battery-powered devices. UWB may also struggle in non-line-of-sight (NLoS) conditions, with performance that is not necessarily better than Wi-Fi or BLE under these circumstances. BLE technology has been researched and developed to address these challenges. Its advantages include low production cost energy efficiency, and easy deployment.

Numerous indoor positioning methods have been developed, each contributing unique advantages to indoor localization. According to Syazwani et al [2], triangulation and trilateration are characterized by low-cost implementation and high accuracy, particularly within room-scale environments. However, both methods can be complex and require angle measurements, leading to reduced accuracy in larger areas and under variable environmental conditions. Proximity offers high accuracy but entails high implementation costs and complexity. Scene analysis demonstrates superior performance but is similarly expensive and complicated. Lastly, fingerprinting is advantageous as it does not necessitate infrastructure; however, its accuracy heavily depends on the resolution of the fingerprinting data, which can complicate data collection and processing. Despite these challenges, fingerprinting is recognized for its relatively high accuracy, making it a compelling choice for indoor positioning. Consequently, this research is centered on developing an indoor positioning method utilizing BLE fingerprinting.

This approach places some Bluetooth beacons (BC) at Predetermined locations. After a predefined period of time, these BCs transmit data packets containing IDs and additional information. The device to be located will continuously collect information and transmit it to the server for processing. The device's location will be estimated based on BLE fingerprint characteristics. This method is divided into two main phases: offline and online. The offline phase collects Received Signal Strength Indicator (RSSI) values from BCs at each reference point (RP). These values are processed to cextract features and stored in a fingerprint map database. The online phase consists of collecting RSSI values from BC signal packets. These values are also used to extract fingerprint features, which are then compared with reference points. The reference points with the most similar fingerprint features are selected to calculate the coordinates of the target location.

This paper proposes an indoor localization method based on BLE fingerprinting, specifically fingerprint feature extraction. It involves deploying six BCs around a room, with the RSSI values of each reference point stored in the fingerprint database. RSSI measurements are susceptible to noise, so the Kalman filter and deep learning models like Autoencoders and Convolutional Autoencoders are employed to reduce noise and data dimensionality. The Minkowski distance is calculated between the measured fingerprint and reference fingerprint to identify the k nearest reference points with the measured fingerprint. This information is used to calculate coordinates and assess accuracy.

This paper is organized as follow: Section 2 presents related research on indoor positioning. Section 3 provides an overview of the dataset construction, and the algorithms used. Experimental results are presented and compared with previous research findings in Section 4. Conclusions and future directions are discussed in Section 5.

## 2. Related work

Before introducing our proposed method, we examine various indoor positioning technologies. There is extensive global research on indoor positioning technology, with methods and technologies outlined in [7]. Some non-object-based positioning technologies are mentioned, such as using cameras for detection and location. Object-based positioning technologies include Bluetooth, Wi-Fi, RFID, Ultra-Wideband, or wireless sensor network technologies. Several articles on indoor positioning methods are summarized in Table 1.

Table 1. Summary of some indoor localization methods.

| Studies | Method | Technique | Accuracy |
|---|---|---|---|
| Shuang Li et al., 2021 | Camera | Use the image and feed it into the detection algorithm | $58\% \leq 0.5m, 77\% \leq 1m$ $95\% \leq 2m,$ $0.61m \sim 9.7m \times 7.8m$ |
| Liang Ma et al., 2019 | RFID tags | Use phase and RSSI signals to feed into POS algorithm | $0.12m \times 1.14m \sim 11m \times 9.3m \times 3m$ |
| Weipeng et al., 2018 | Wireless Local Area Network | Use existing WLAN infrastructure (APs) | $\sim 1-3\ m$ |

| | | | |
|---|---|---|---|
| Samaneh et al., 2017 | Wifi Fingerprint | KNN algorthm | $0.83m \sim 8m \times 5{,}25\ m$ |
| Ashry et al., 2019 | Wifi | Trilateration and fingerprinting methods | $2.8\ m \sim 2.5m \times 85\ m$ |
| Mai et al., 2020 | Bluetooth Fingerprint | Pedestrian Dead Reckoning + Fingerprinting + Particle filter | $1.18m \sim 35.25\ m^2$ |
| Alvin Riady et al., 2022 | Bluetooth Fingerprint | ANN | $1.1178m \sim 19m \times 12m$ |

Several BLE fingerprinting methods have been proposed. Zou and colleagues [3] applied graph optimization to achieve a best-case accuracy of 1.27 meters. Martin and colleagues [4] employed Gaussian kernel-based fingerprinting with an accuracy below 1.5 meters in 90% of cases. Subedi and colleagues [5] utilized a two-step fingerprint-based approach with an accuracy of 1.05 meters. Li and colleagues [6] utilized an eight-neighborhood template-matching mechanism with a 1-meter accuracy.

The study [7] has identified the most suitable technologies for Indoor Positioning Systems (IPS) and their principles, methods, and algorithms. The algorithm that uses the phone camera to take images and produce algorithmic findings has been mentioned by Shuang Li et al [8]. However, the results showed a 95% margin of error greater than 2 meters, with an average positioning error of 0.61 meters.

Other algorithms utilize RFID tags, as demonstrated in [8]. Liang Ma et al [8] employed phase and RSSI signals in their Positioning (POS) algorithm for location calculations. Most non-invasive positioning technologies rely on infrastructure, such as wireless sensor networks. These sensors detect changes when people or objects move within the area. For instance, in [9], the authors used a wireless local area network to model their positioning system.

Nonetheless, non-invasive positioning technologies relying on infrastructure can significantly increase system construction costs and may need to be more versatile across different environments. On the other hand, technologies such as Wi-Fi, Bluetooth, and Ultra-Wideband enable user positioning through the devices they carry. Indoor wireless positioning systems use RSSI signals to determine coordinates, as discussed in [10] and [11]. Samaneh et al. [10] employed Wi-Fi RSSI signals to create fingerprints, which were then utilized for training and testing. Various algorithms, such as neural networks, KNN, and SVM, can employ fingerprints. Typically, they compute the correlation between RSSI signals during positioning and pre-existing fingerprint datasets to provide the closest location. In [11], the trilateration method and Wi-Fi RSSI fingerprint were used for positioning. [12] combined fingerprint and Pedestrian Dead Reckoning with the Particle Filter method to infer coordinates from signal transmission distances. Alvin et al. [13] also employed fingerprints using the ANN method. Models using physical objects for positioning offer higher accuracy but increase computational complexity and processing time. Fingerprinting has gained traction recently with the use of neural networks, KNN, SVM, and Euclidean distance. However, these studies have yet to address cost and energy efficiency concerns effectively. Therefore, this research proposes a low-energy Bluetooth fingerprinting method to address these issues while maintaining the highest possible accuracy.

The paper is organized as follows. Section 3 describes in detail the algorithms used in this paper. The experiment results are shown in section 4. The conclusion is given in section 5.

## 3. Method

The critical steps of indoor positioning using fingerprint features are illustrated in Figure 1. Firstly, specific Reference Points (RPs) are determined in the positioning area, and the RSSI values of BLE beacons are measured at each RP. These RSSI values are then stored in the fingerprint database. Subsequently, the RSSI values are passed through a Kalman filter to eliminate noise while preserving stability in the RSSI values at each reference point. Afterward, Autoencoders (AE) or Convolutional Autoencoders (CAE) models extract fingerprint features at each reference point. Finally, a normalized cross-correlation method compares the correlation between the point to be located and all the reference points in the database. The coordinates of the points to be determined are calculated as the average of the k points with the highest correlation. Each step in our proposed method is presented in detail below.

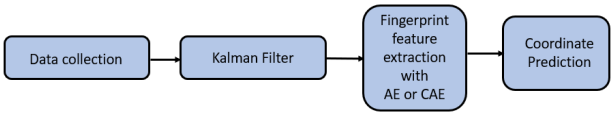


**Figure 1.** The main steps in the proposed method

## 3.1. Data collection

Let's assume that there are $N$ reference points within the coverage area. At each reference point, RSSI values are measured over a period and organized into the following matrix:

$$R = \begin{bmatrix} r_1(1) & r_1(2) & \cdots & r_1(B) \\ r_2(1) & r_2(2) & \cdots & r_1(B) \\ \vdots & \vdots & \ddots & \vdots \\ r_N(1) & r_N(2) & \dots & r_N(B) \end{bmatrix} \quad (1)$$

In matrix (1), $r_n(b)$ represents the RSSI value at reference point $n$ obtained from beacon $b$. Here, $n = 1, 2, 3, \dots N$, representing the sequential number of reference points, and $b = 1, 2, 3, \dots B$, the number of beacons used within a defined range.

## 3.2. Kalman Filter

The Kalman filter, introduced by Rudolf E. Kalman and published in 1960 [14] is a widely used tool in control systems. It is employed to estimate the state of a process in the presence of noise in measurements. This method works by determining the estimated state of the process based on actual measurements and the ideal state, to minimize the mean square error between them. The Kalman filter consists of two primary steps: *Prediction* and *Measurement Update* [15], [16]. The visualization of the Kalman filter process is depicted in Figure 2.

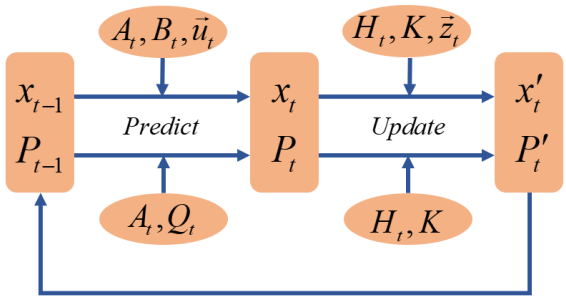


**Figure 2.** Implementation of the Kalman filter.

**Prediction**

The current state $x_t$ and error covariance matrix $P_t$ of the process are estimated in a general form as:

$$x_t = A_t x_{t-1} + B_t u_t \quad (2)$$
$$P_t = A_t P_{t-1} A_t^T + Q \quad (3)$$

Where:

$A_t$: State transition model matrix
$B_t$: Control input model matrix
$u_t$: Control vector
$Q_t$: Process noise covariance matrix

**Measurement Update**

The initial task in the update process is to compute the Kalman Gain, as shown in Eq.4:

$$K = P_t H_t^T (H_t P_t H_t^T + R_t)^{-1} \quad (4)$$

Next, the expected state and covariance matrix are updated as per Eq.5 and Eq.6:

$$x'_t = x_t + K(z_t - H_t x_t) \quad (5)$$
$$P'_t = (1 - K_t H_t) P_t \quad (6)$$

where, $H_t$ is the matrix relating to state $x_t$ through the measurement $z_t = H_t x_t + R_t$, where $R_t$ is a random variable representing the measurement noise covariance. The Kalman filter operates recursively: the *Prediction* process estimates the current provisional state based on the previous state, and then the *Measurement Update* process adjusts the estimate with an actual measurement. These steps are repeated with previous posterior estimates used to predict new prior estimates [16].

With our collected RSSI data, each vector $r_n(k) = \{rssi_1, rssi_2, \dots, rssi_R\}$ is passed through the Kalman filter, with the first value as the average of R samples in each vector:

$$rssi_0 = \frac{1}{R} \sum_{i=1}^{R} rssi_i \quad (7)$$

The Kalman filter enhances the stability of our dataset, thereby improving the fingerprint features for each reference point and enhancing training performance.

## 3.3. Fingerprint Features Extraction

### 3.3.1. Autoencoder

Autoencoder (AE) is a neural network model in machine learning and computer vision designed for unsupervised data encoding. It aims to learn a lower-dimensional representation (encoding) for higher-dimensional data, reducing complexity and saving computational resources. AE is often used for dimensionality reduction and feature extraction tasks. Figure 3 provides a visual representation of AE architecture, consisting of *Encoder*, *Code*, and *Decoder*.

*Encoder*: Receives input data and transforms it into a lower – dimensional compressed form. The encoder typically consists of a sequence of neuron layers, learning to extract essential information from the data and represent it as a compressed vector. The neuron layers in the encoder often employ activation functions like ReLU, sigmoid, or hyperbolic tangent.

*Code*: Contains the compressed data, also known as the output of the encoder. It is a crucial part of the network because it holds the features of the input data.

*Decoder*: Receives the compressed data from the encoder and attempts to reconstruct the original data. The decoder also consists of a sequence of neuron layers, transforming the compressed data into the original data while minimizing the reconstruction error.

The training process of an Autoencoder aims to minimize the error between the original data and the reconstructed data by adjusting the encoder and decoder weights and parameters. Loss functions commonly include Mean Squared Error (MSE) and Binary Cross-Entropy (BCE).

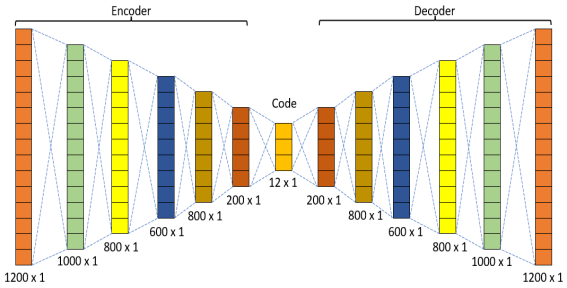


**Figure 3.** Proposed Autoencoder model structure

Each reference point in our database has data vectors of size $200 \times 6$, which are flattened into $1200 \times 1$ vectors to match the input size of the AE model. After passing through the encoder, the data is compressed into a $12 \times 1$ code, which is then decoded to produce an output of $1200 \times 1$. In this study, the Autoencoder model uses the hyperbolic tangent (tanh) activation function, employs the Adam optimization algorithm, and uses Mean Squared Error (MSE) as the loss function.

### 3.3.2. Convolutional Autoencoder

The Convolutional Autoencoder (CAE) combines convolutional neural network principles with an autoencoder. It is often used for unsupervised learning tasks. Like an autoencoder, the CAE architecture consists of an Encoder, Code, and Decoder [17]. The proposed CAE architecture in this study is illustrated in Figure 4.

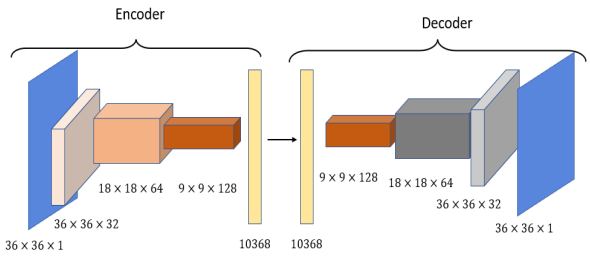


**Figure 4.** Proposed Convolutional Autoencoder model structure.

The encoding part processes the input as a matrix using convolutional layers to produce lower-dimensional output than the input matrix. The decoding part takes the lower-dimensional representation from the encoding part and transforms it back to the original matrix size using decoding layers. The training process of the Convolutional Autoencoder is similar to that of the Autoencoder, with the aim of minimizing the difference using Mean Squared Error (MSE) as the loss function.

Because the input of the CAE model is a matrix, each $1200 \times 1$ data vector is transformed into a $36 \times 36$ matrix with zero-padding elements.

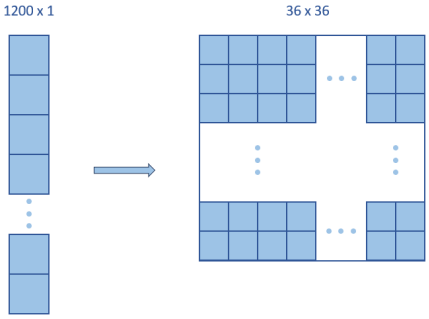


**Figure 5.** The data vector is converted to matrix form for input to the Convolutional Autoencoder.

## 3.4. Coordinate Prediction

### 3.4.1. Correlation

Signal correlation is a crucial aspect in signal research and analysis. In this study, a correlation system is used to compute and compare the input signal with an available fingerprint dataset. For two discrete signals $x[n]$ and $y[n]$, the calculation of their correlation, denoted as $C(x, y)$, is performed using the following formula:

$$C(x,y) = \sum_{n_1}^{n_2} x[n]y[n] \tag{8}$$

Where $n_1$ and $n_2$ represent specific time intervals for calculating the correlation between the two signals [19]. In a special case where the two signals are identical, it can be observed that in this case, the main correlation is the signal's energy:

$$C(x,x) = E(x) \tag{9}$$

### 3.4.2. Normalized Cross – Correlation

Normalized Cross-Correlation (NCC) is used in signal processing to measure the degree of similarity or correlation between two signals. NCC is typically employed to search for a specific signal pattern within a larger signal.

This research proposes using the NCC coefficient to compare the input signal with a pre-existing fingerprint database to determine the most accurate coordinates. NCC

between two signals $x[n]$ and $y[n]$ is determined by the following formula (10):

$$NCC(x,y) = \frac{\sum_{n_1}^{n_2} x[n]y[n]}{E(x)E(y)} \quad (10)$$

This formula normalizes the aggregate correlation by dividing the numerator by the product of the energy of two signals, $x[n]$ and $y[n]$. The result falls within the range of $-1$ to 1, indicating the level of similarity between the two signals. A value of 1 typically represents complete correlation, while $-1$ indicates complete inverse correlation. A value close to 0 generally indicates low or no correlation between the two signals.

Utilizing this approach involves the identification of $k$ reference points exhibiting the closest distance. Eventually, the point coordinates to be determined are predicted as the centroid of these $k$ reference points. Different values of $k$ result in different predicted coordinates, calculated using formula (11):

$$(x,y) = \frac{1}{k}\sum_{i=1}^{k}(x_i, y_i) \quad (11)$$

## 4. Experiments and Results

### 4.1. Data collection

The experiment was conducted on the 6th floor of the Ta Quang Buu Library at Hanoi University of Science and Technology, Vietnam. Six BLE beacons were placed at coordinates $(0,0), (0,4), (0,8), (8,0), (8,4)$, and $(8,0)$ within an $8m \times 8m$ area, as described in Figures 6 and 7. The beacons and Bluetooth signal strength receiving devices were on the same floor of the plane.

There is a total of 75 reference points on the map. At each reference point, 200 RSSI value samples were gathered for a specific beacon. Additionally, 20 random test points were collected to assess the performance of the fingerprint feature extraction model, as depicted in Figure 7 and Figure 8.

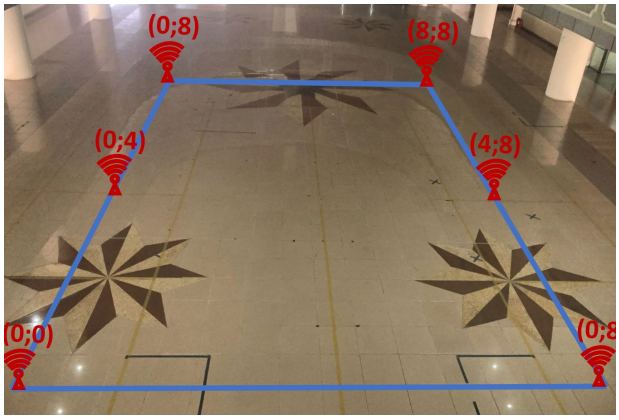


**Figure 6.** The experimental environment.

Figure 9 presents a specific example of RSSI data from 200 samples recorded at two reference points ($RP_1$ and $RP_2$) fo r a sp ecific be acon. Co nversely, Fi gure 10 illustrates RSSI data from 200 samples obtained from two different beacons ($BC_1$ and $BC_2$) at a reference point. These data illustrate the uneven signal variations. This inconsistency may be due to the influence of the surrounding environment and factors causing random errors during the experimental process. This issue poses a significant challenge for indoor localization methods relying on BLE signals.

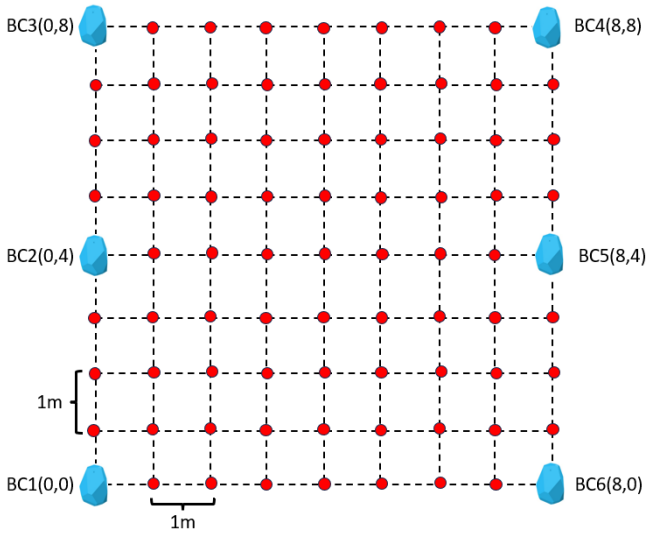


**Figure 7.** Arrange the experiment to collect RSSI values from beacons at each reference point.

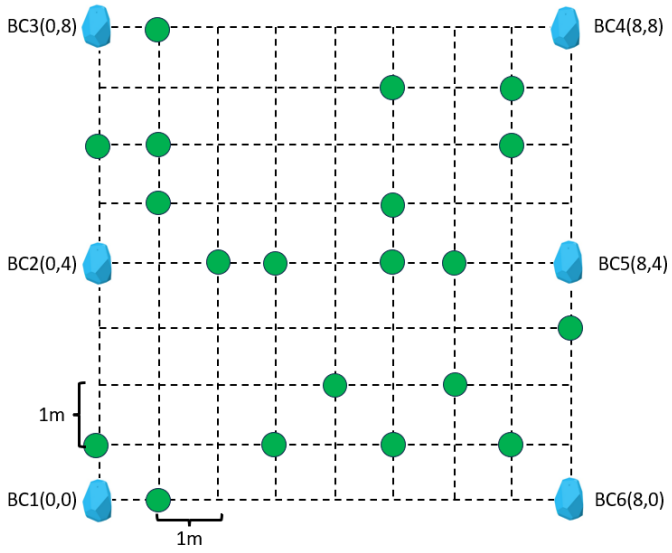


**Figure 8.** Test points are collected randomly.

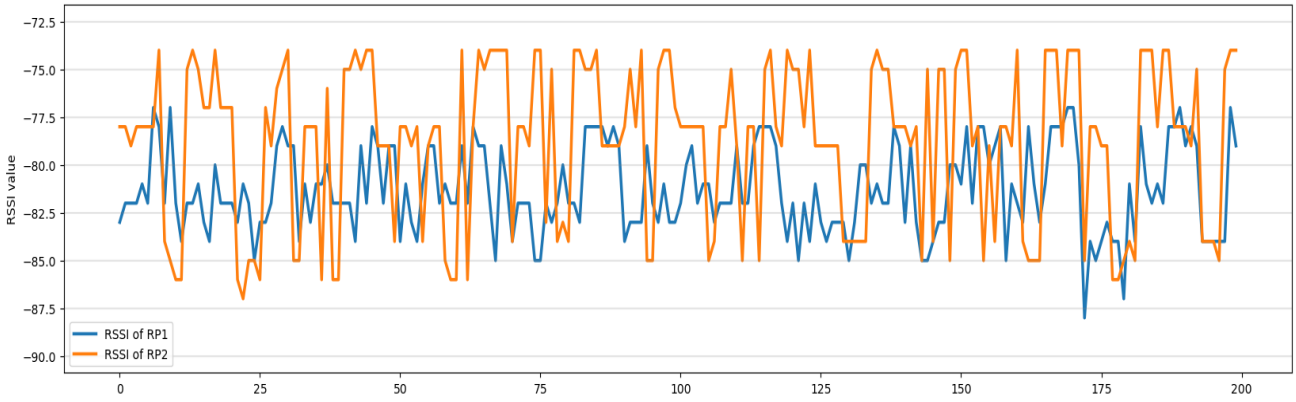


**Figure 9.** The RSSI values of the same beacon is obtained at different reference point.

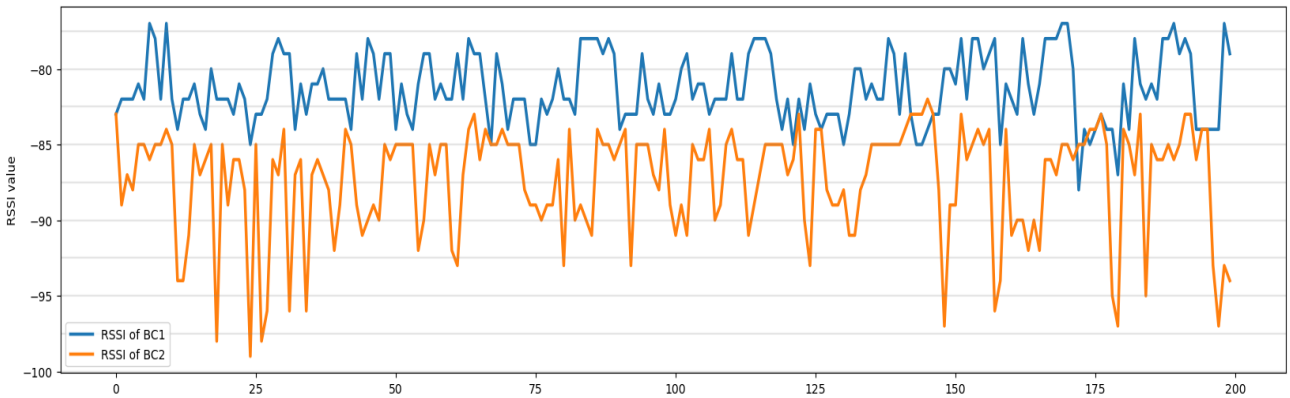


**Figure 10.** The RSSI values obtained from two different beacons at the same reference point.

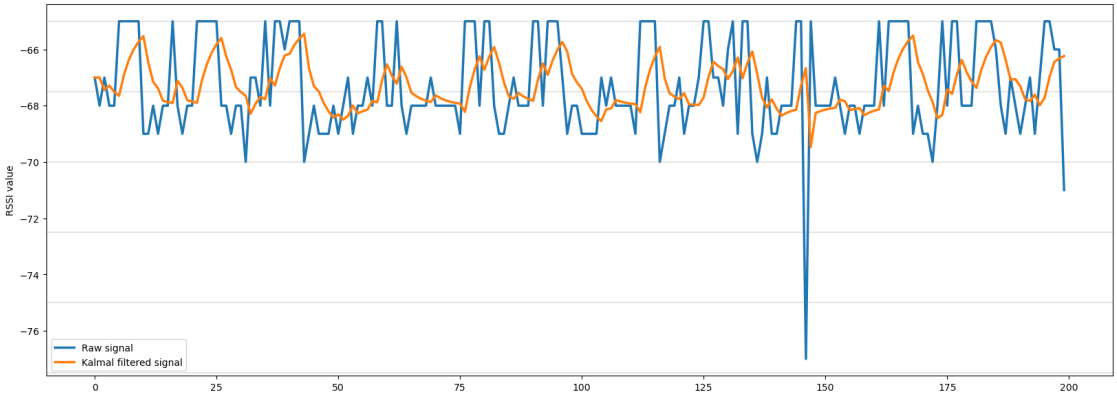


**Figure 11.** The raw RSSI values and after passing it through the Kalman Filter.

## 4.2. Utilizing Kalman Filter

As explained in the previous section, noise factors can significantly affect the process of fingerprint feature extraction and BLE signal-based localization. Consequently, the collected database underwent Kalman filtering to partially reduce the noise in the aforementioned values. Moreover, it enhances the feature characteristics of RSSI values at each reference point. Figure 11 below illustrates the difference before and after employing Kalman filtering. It is evident that, after passing through the Kalman filter, the RSSI data eliminates noisy values, resulting in new, more stable data.

## 4.3. Experimental results

Table 2 displays the results using two methods, one incorporating the Kalman filter and the other without the filter, with different values of $k$ ($k = 3,4,5,6,7$). Parameters in the table include the mean, median, maximum, and minimum error values. Initially, a comparison is made between two methods, AE and CAE, revealing that the average error values for different k values are notably smaller with the CAE method than with the AE method. The AE method provides the smallest average error of $2.60m$ with $k = 4$, while the CAE method yields the smallest average error of $1.07m$ with the same k value. When combined with the Kalman filter, it can be observed that the accuracy of the localization task improves. Specifically, with $k = 4$, the AE-Kalman method achieves the smallest average error of $1.16m$, which is an improvement compared to AE ($2.6m$), and the CAE-Kalman method delivers the smallest average error of $0.98m$ compared to CAE's $1.07m$. Of the four experimented methods, CAE-Kalman demonstrated the highest stability, showcasing a localization error ranging from $0.12m$ ($k = 4$) to $2.39m$ ($k = 6$). In contrast, the AE method shows a maximum localization error of $5.49m$ at $k = 3$.

Figures 12 illustrate cumulative distribution function (CDF) curves for the following methods: CAE-Kalman, CAE, AE-Kalman, and AE. It can be observed that CAE-Kalman and CAE have similar CDF curves, while AE-Kalman and AE exhibit similar curves. When the Kalman filter is applied, CAE-Kalman outperforms CAE, and AE-Kalman outperforms AE. CAE-Kalman and CAE have a lower error range of less than $2m$, whereas AE-Kalman and AE have an error range of less than $4m$. Using Kalman filtering improves the performance of fingerprint-based localization and reduces the error range.

Figure 13 illustrates the localization error box and whisker plots for the four methods employed in this study, specifically at a value of $k = 4$. It is evident that the CAE method generally outperforms the AE method, and the comparison between applying and not applying the Kalman filter in data processing shows a clear difference in efficiency.

The CAE method combined with the Kalman filter ($k = 4$) is compared to studies using native BLE fingerprint-based localization in Table 3. The comparison is made on various aspects such as the number of beacons, used, the area size, and the minimum, average, and maximum, location errors. As explained in section 2, Mai et al [12] used fingerprinting combined with Pedestrian Dead Reckoning and Particle filter to achieve a minimum average error of $1.18m$. Alvin Riady et al [13], with a larger localization scale and a greater number of beacons than our method, achieved minimum average and maximum errors of $1.1178m$ and $3.3601m$, respectively. Li et al [6] used the ENTM method, developed by the KNN and WKNN methods, and achieved an average error of 1m. Table 4 compares the CAE method combined with the Kalman filter and other methods. The comparison table shows that the proposed CAE method combined with the Kalman filter achieved an average error of $0.98m$, significantly outperforming the compared methods.

Table 2. Statistical parameters of the proposed methods with different k values (unit: m).

| *Methods* | *Statistics* | *k = 3* | *k = 4* | *k = 5* | *k = 6* | *k = 7* |
|---|---|---|---|---|---|---|
| *CAE-Kalman* | *Mean* | *1.19* | *0.98* | *1.24* | *1.37* | *1.41* |
| | *Min* | *0.33* | *0.12* | *0.20* | *0.37* | *0.52* |
| | *Max* | *2.33* | *1.60* | *2.24* | *2.39* | *2.33* |
| | *Median* | *1.05* | *1.02* | *1.22* | *1.20* | *1.42* |
| | *Var* | *0.31* | *0.21* | *0.28* | *0.26* | *0.26* |
| | *Std* | *0.55* | *0.46* | *0.53* | *0.51* | *0.51* |
| *CAE* | *Mean* | *1.12* | *1.07* | *1.21* | *1.18* | *1.37* |

| *Methods* | *Statistics* | *k = 3* | *k = 4* | *k = 5* | *k = 6* | *k = 7* |
|---|---|---|---|---|---|---|
| | *Min* | *0.23* | *0.2* | *0.27* | *0.33* | *0.40* |
| | *Max* | *2.57* | *2.15* | *2.41* | *2.27* | *2.59* |
| | *Median* | *1.05* | *1.02* | *1.22* | *1.12* | *1.25* |
| | *Var* | *0.41* | *0.28* | *0.32* | *0.28* | *0.46* |
| | *Std* | *0.64* | *0.53* | *0.56* | *0.53* | *0.68* |
| *AE-Kalman* | *Mean* | *1.25* | *1.16* | *1.21* | *1.37* | *1.46* |
| | *Min* | *0.33* | *0.25* | *0.28* | *0.47* | *0.87* |
| | *Max* | *3.07* | *2.85* | *2.61* | *2.69* | *3.03* |
| | *Median* | *1.05* | *1.09* | *1.09* | *1.31* | *1.38* |
| | *Var* | *0.44* | *0.29* | *0.25* | *0.24* | *0.26* |
| | *Std* | *0.66* | *0.54* | *0.50* | *0.49* | *0.51* |
| *AE* | *Mean* | *2.82* | *2.60* | *2.76* | *2.69* | *2.72* |
| | *Min* | *0.94* | *0.25* | *0.28* | *0.17* | *0.77* |
| | *Max* | *5.49* | *4.75* | *5.07* | *4.78* | *4.90* |
| | *Median* | *2.96* | *2.65* | *2.66* | *2.71* | *2.71* |
| | *Var* | *1.94* | *1.51* | *1.95* | *1.70* | *1.64* |
| | *Std* | *1.39* | *1.23* | *1.40* | *1.31* | *1.28* |

Table 3. Comparison of CAE incorporating Kalman filter with other fingerprint-based methods

| **Studies** | **Methods** | **Number of Becons** | **Area size (m x m)** | *Minimum Error* **(m)** | *Average Error* **(m)** | *Maximum Error* **(m)** |
|---|---|---|---|---|---|---|
| Mai et al [12] | Pedestrian Dead Reckoning + Fingerprinting + Particle filter | 8 | 35,25 | --- | 1.18 | --- |
| Alvin Riady et al [13] | ANN | 23 | 19 × 12 | 0.1055 | 1.1178 | 3.3601 |
| Mingfeng Li et al [6] | ENTM | 4 | 8 × 8 | --- | 1 | --- |
| This study | CAE + Kalmam filter | 6 | 8 × 8 | 0.10 | 0.98 | 1.77 |

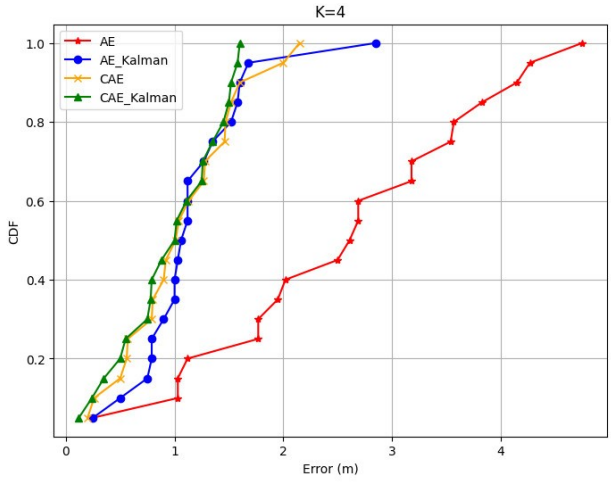


**Figure 12.** Comparison of the localization error CDF curves of four methods, where $k = 4$

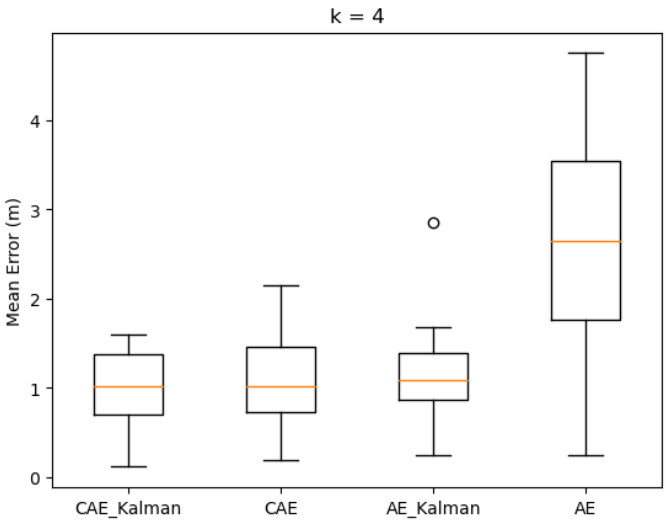


**Figure 13.** The localization error CDF curves box-whisper plots, where $k = 4$

## 5. Conclusion

In this research, we employed four distinct methods to assess the performance of indoor localization: CAE_Kalman, CAE, AE_Kalman, and AE. Our study results reveal that the CAE model outperforms the AE model, highlighting the superiority of the CAE model in fingerprint feature extraction for indoor localization. Additionally, we examined the impact of applying the Kalman filter to both models. The results demonstrate that using the Kalman filter significantly enhances the performance of both models compared to not using the filter. This underscores the effectiveness of improving the stability and accuracy of RSSI values obtained from BLE beacon signal transmitters. In summary, this research has elucidated the excellence of the CAE model and the positive effects of the Kalman filter in enhancing the performance of fingerprint feature extraction for indoor localization.

## References

[1] Leitch, S. G., Ahmed, Q. Z., Abbas, W. B., Hafeez, M., Laziridis, P. I., Sureephong, P., & Alade, T. (2023). On indoor localization using WiFi, BLE, UWB, and IMU technologies. *Sensors*, *23*(20), 8598. https://doi.org/10.3390/s23208598

[2] J, N. S. C., Wahab, N. H. A., Sunar, N., Ariffin, S. H. S., Wong, K. Y., & Aun, Y. (2022). Indoor Positioning System: A Review. *International Journal of Advanced Computer Science and Applications*, *13*(6). https://doi.org/10.14569/ijacsa.2022.0130659

[3] Z. Zuo, L. Liu, L. Zhang, and Y. Fang, "Indoor Positioning Based on Bluetooth Low-Energy Beacons Adopting Graph Optimization," Sensors, vol. 18, no. 11, Art. no. 11, Nov. 2018, doi: 10.3390/s18113736.

[4] P. Martins, M. Abbasi, F. Sa, J. Celiclio, F. Morgado, and F. Caldeira, "Intelligent beacon location and fingerprinting," Procedia Comput. Sci., vol. 151, pp. 9–16, Jan. 2019, doi: 10.1016/j.procs.2019.04.005.

[5] S. Subedi, H.-S. Gang, N. Y. Ko, S.-S. Hwang, and J.-Y. Pyun, "Improving Indoor Fingerprinting Positioning With Affinity Propagation Clustering and Weighted Centroid Fingerprint," IEEE Access, vol. 7, pp. 31738–31750, 2019, doi: 10.1109/ACCESS.2019.2902564.

[6] M. Li, L. Zhao, D. Tan, and X. Tong, "BLE Fingerprint Indoor Localization Algorithm Based on Eight-Neighborhood Template Matching," Sensors, vol. 19, no. 22, p. 4859, Nov. 2019, doi: 10.3390/s19224859.

[7] L. Batistić and M. Tomic, "Overview of indoor positioning system technologies," in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), May 2018, pp. 0473–0478. doi: 10.23919/MIPRO.2018.8400090.

[8] L. Ma, M. Liu, H. Wang, Y. Yang, N. Wang, and Y. Zhang, "WallSense: Device-Free Indoor Localization Using Wall-Mounted UHF RFID Tags," Sensors, vol. 19, no. 1, Art. no. 1, Jan. 2019, doi: 10.3390/s19010068.

[9] W. Guan, S. Wen, H. Zhang, and L. Liu, "A Novel Three-dimensional Indoor Localization Algorithm Based on Visual Visible Light Communication Using Single LED," in 2018 IEEE International Conference on Automation, Electronics and Electrical Engineering (AUTEEE), Nov. 2018, pp. 202–208. doi: 10.1109/AUTEEE.2018.8720798.

[10] S. Amirisoori, S. Mohd Daud, N. A. Ahmad, S. Natasha, N. Sa'at, and N. Noor, "WI-FI Based Indoor Positioning Using Fingerprinting Methods (KNN Algorithm) in Real Environment," Int. J. Future Gener. Commun. Netw., vol. 10, pp. 23–36, Sep. 2017, doi: 10.14257/ijfgcn.2017.10.9.03.

[11] A. E. M. E. Ashry and B. I. Sheta, "Wi-Fi based indoor localization using trilateration and fingerprinting methods," IOP Conf. Ser. Mater. Sci. Eng., vol. 610, no. 1, p. 012072, Sep. 2019, doi: 10.1088/1757-899X/610/1/012072.

[12] T.-M. T. Dinh, N.-S. Duong, and K. Sandrasegaran, "Smartphone-Based Indoor Positioning Using BLE iBeacon

and Reliable Lightweight Fingerprint Map," IEEE Sens. J., vol. 20, no. 17, pp. 10283–10294, Sep. 2020, doi: 10.1109/JSEN.2020.2989411.
[13] A. Riady and G. P. Kusuma, "Indoor positioning system using hybrid method of fingerprinting and pedestrian dead reckoning," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 9, pp. 7101–7110, Oct. 2022, doi: 10.1016/j.jksuci.2021.09.005.
[14] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," J. Basic Eng., vol. 82, no. 1, pp. 35–45, Mar. 1960, doi: 10.1115/1.3662552.
[15] B. Alsadik, "Chapter 10 - Kalman Filter," in Adjustment Models in 3D Geomatics and Computational Geophysics, vol. 4, B. Alsadik, Ed., in Computational Geophysics, vol. 4. , Elsevier, 2019, pp. 299–326. doi: 10.1016/B978-0-12-817588-0.00010-6.
[16] M. George and K. Vadivukkarasi, "Kalman filtering for RSSI based localization system in wireless sensor networks," vol. 10, pp. 16429–16440, Jan. 2015.
[17] A. Kargar-Barzi, E. Farahmand, A. Mahani, and M. Shafique, "CAE-CNNLoc: An Edge-based WiFi Fingerprinting Indoor Localization Using Convolutional Neural Network and Convolutional Auto-Encoder".
[18] E. Pintelas, I. E. Livieris, and P. E. Pintelas, "A Convolutional Autoencoder Topology for Classification in High-Dimensional Noisy Image Datasets," Sensors, vol. 21, no. 22, p. 7731, Nov. 2021, doi: 10.3390/s21227731.
[19] "EECS 206 Labs." Accessed: Oct. 09, 2023. [Online]. Available: https://www.eecs.umich.edu/courses/eecs206/public/la