

Deep Learning-Based Bluetooth Low-Energy 5.1 Multianchor Indoor Positioning with Attentional Data Filtering

Zhongyuan Lyu,* Tom Tak-Lam Chan, Theo Yik-Tung Hung, Hang Ji, Gary Leung, and Daniel Pak-Kong Lun*

Indoor positioning system (IPS) technologies have widespread applications in logistics, intelligent manufacturing, healthcare monitoring, etc. The recently released Bluetooth low-energy (BLE) 5.1 specification enables in-phase and quadrature-phase (I/Q) data measurements. It allows angle of arrival estimation and becomes a natural choice for IPS implementation. Conventional BLE 5.1 IPSs use multiple anchors to provide massive redundancy to improve system robustness. It however demands effective approaches to leverage redundancy. Besides, interference due to various environmental factors can introduce severe errors to I/Q data and affect positioning accuracy. Facing these challenges, herein, a novel deep learning-based multianchor BLE 5.1 IPS is proposed. The system aggregates measurements from multiple anchors and makes them available at regular time steps. Then, a novel attentional filtering network tailored to infer high-quality I/Q sample data is developed and a spatial regularization loss incorporating spatial location relationships to strengthen the feature embedding discrimination is proposed. Two multianchor BLE 5.1 I/Q sample datasets are developed and released for public download. Numerical experiments are carried out to compare the proposed method with previous BLE 5.1 IPS methods and methods utilizing other radio frequency data. Results indicate that the proposed method consistently achieves submeter accuracy and significantly outperforms the state-of-the-art approaches.

positioning systems based on radio frequency (RF) signals has attracted significant attention from researchers and industrial practitioners as a challenging and important task.^[3]


Various wireless techniques have been investigated and deployed for localization in indoor environments.^[4–6] Among all the techniques, Wi-Fi-based localization is predominantly adopted in most IPS applications^[2] as it can be deployed without adding new infrastructure. Abundant information can be obtained from Wi-Fi signals, such as Channel State Information (CSI), received signal strength (RSS) value, and media access control (MAC) address of access points (APs), to develop geometry-based and data-driven positioning algorithms. However, most Wi-Fi APs require AC power. Installing a new Wi-Fi AP can be expensive due to the high cabling cost. Besides, it will be difficult to implement Wi-Fi-based IPS in venues with no AC power access. Recently, the new bluetooth low-energy (BLE) 5.1 standard was released in 2019. It includes the direction-finding features named constant tone extension (CTE)

for IPS.^[7] Similar to the previous Bluetooth systems, the BLE 5.1 systems improve over the Wi-Fi-based systems by their low power consumption. By switching among antennas, BLE receivers can measure the in-phase and quadrature-phase (I/Q) values from the CTE of BLE packets and estimate the positions of the transmitters.^[8]

1. Introduction

Indoor positioning systems (IPS) play a crucial role in indoor location-based service and have shown their immense potential market values in different fields such as indoor navigation and healthcare monitoring.^[1,2] Developing efficient and accurate

Z. Y. Lyu, T. T. L. Chan, T. Y. T. Hung, D. P. K. Lun
Centre for Advances in Reliability and Safety
Hong Kong 999077, China
E-mail: zhongyuan.lyu@cairs.hk

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300292>.

© 2023 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300292

H. Ji, D. P. K. Lun
Department of Electrical and Electronic Engineering
The Hong Kong Polytechnic University
Hong Kong 999077, China
E-mail: enpkun@polyu.edu.hk

G. Leung
Blue Pin (HK) Limited
Hong Kong 999077, China

While deep neural networks are successfully applied to different application domains, it is no surprise that they are also used in IPS design.^[9–11] Recently, some efforts have been made to develop deep learning (DL)-based models for BLE 5.1 IPS such as using the convolutional neural network (CNN)^[11] and recurrent neural network.^[10] These approaches often use the fingerprint-based method such that object positioning is converted into a classification problem of the I/Q sample fingerprints. Despite the merits of previous research, there are still some critical challenges in the integration of DL models with the positioning process with multianchor BLE 5.1 I/Q sample data. First, I/Q sample data is easily disturbed by interference (such as object blocking, people movement, and the existence of other RF signals) in the environment. The distorted data can affect the constructed fingerprints and result in significant errors in the estimation.^[12] Some researchers have investigated fitting methods for I/Q sample data.^[13] They developed a phase difference density-based classifier (PDDC) to choose the dominant cluster of phase difference values calculated from I/Q samples. In other studies,^[12,14] the authors used the Kalman filter to deal with the noise and other errors in the measurements. However, the filtering parameters are generally fixed for all locations.^[15] On the one hand, different locations may have different optimal filtering parameters (pointwise setting), which are hard to set without (prior) knowledge of the indoor environment. On the other hand, current filtering methods for I/Q sample data do not well integrate into the DL-based IPS. Rather than a separate filtering process, an integrated filtering network that can be end-to-end trained with the position estimation network will be more effective for both processes to approach optimality.

Second, the existing fingerprint-based solutions are developed mainly based on the phase difference and RSS,^[9,13] which are usually used for geometry-based algorithms. The I/Q sample data calculated from CTE based on multiantenna switching is not fully utilized. With the various spatial arrangements of antenna architectures (linear, rectangular, circular) and complex indoor environments, I/Q data sampled by each antenna can have different signal gains. Such spatial correlations help construct unique features. Besides, previous BLE 5.1 IPS research assumes that the data are collected simultaneously from all anchors/broadcasting channels.^[9,10] It is indeed not the case in practice. Anchor data present themselves randomly within a time period. It is possible to have multiple data received from a particular channel of one anchor in the period. It is also possible that no data is received from an anchor in the period due to interferences or other hardware-related problems.^[16] The irregular arrival time of the anchor data introduces much difficulty when using them to construct input features for the subsequent deep neural network. The utilization of data packets from multiple anchors for positioning has not been well investigated in previous research.

Facing these challenges, this research develops a new DL-based multianchor BLE 5.1 IPS and a novel positioning network model named attentional filtering indoor positioning network (AnFIPNet). To fully utilize the correlation in the I/Q sample data, we extract both the phase difference and amplitude data from the I/Q samples to form the input features. Different from the existing approach which constructs an input feature by collecting data within a fixed time interval, we propose to collect data within a time step in which the amount of data is guaranteed. It reinstalls

the regularity from the irregular arrival times of multiple anchor/multiple channels data. To tackle the measurement error problems in real-time positioning, we develop an attentional filtering network to infer high-quality data and construct features by applying a weighted combination of the data received by each anchor. Low-quality data will be identified and given smaller weights so that they are suppressed in the features. The features are then fed to a position estimation network which contains a convolutional layer and a fully connected (FC) layer to obtain the latent feature representation for each anchor. The feature embeddings are then classified to determine the position of the object. Both the attentional filtering network and position estimation network are trained end to end to approach optimality. To strengthen the discrimination power among feature embeddings, we propose a spatial regularization loss that incorporates the spatial relationships of the I/Q sample data to regularize the position estimation. This new loss function is used together with the cross-entropy loss when training AnFIPNet. Extensive experiments were performed to test the proposed model and other baseline methods on two real-world datasets which we developed particularly for this research. State-of-the-art performance is achieved consistently.

The contribution of this article can be summarized as follows.

- 1) We propose a new multianchor BLE 5.1 indoor positioning system. A new approach is adopted to construct the input features so that the redundancy in the sampled I/Q data can be fully utilized and a steady data flow can be guaranteed to facilitate the smooth operation of the subsequent deep neural network positioning model.
- 2) We propose a novel deep neural network model, namely AnFIPNet, for realtime multi-anchor BLE 5.1 indoor positioning. We integrate data filtering operations into the model by introducing an attentional filtering network to infer high-quality features before feature embedding. We also propose a new spatial regularization loss function for training the model to enhance the discrimination ability of feature embeddings.
- 3) We compare AnFIPNet to previous BLE 5.1 indoor positioning methods and other advanced IPS models based on attention mechanism. We also compare AnFIPNet with IPSs using other RF data such as RSS and Wi-fi CSI. Experiment results show that AnFIPNet can consistently achieve submeter accuracy and achieves the best performance in measurement accuracy compared with all state-of-the-art methods.
- 4) We develop two real-world I/Q sample datasets collected from the developed multianchor BLE 5.1 IPS in different indoor environments. The datasets will be made publicly available.

The remaining part of this article is organized as follows. Section 2 reviews the literature related to indoor positioning, BLE 5.1, and DL-based methods for IPS. Section 3 describes the developed multianchor BLE 5.1 IPS and the details of the proposed AnFIPNet. In Section 4, the construction of the datasets and experimental evaluation results are discussed. We conclude this article in Section 5.

2. Literature Review

2.1. BLE 5.1 AoA in IPS

This subsection focuses on the previous research on the BLE 5.1 angle-of-arrival (AoA) estimation system for indoor positioning.

Toasa et al.^[17] implemented a BLE 5.1 AoA estimation system and discussed the corresponding techniques at the software and hardware level. The experimental results showed that the proposed system could precisely estimate the angle in the lower AoA range (-60° to 60°). Cominelli et al.^[8] evaluated the positioning accuracy using the BLE 5.1 system and revealed the restricted angular detection range of BLE 5.1 anchors. Ye et al.^[18] investigated the BLE AoA system and proposed the corresponding angle estimation algorithm based on signal fitting and propagator direct data acquisition (PDDA). It was shown that the PDDA approach has a lower complexity than the traditional MUSIC algorithm. However, the approach only considers the AoA data from a single anchor, which is challenging when applied to complex indoor environments.

The BLE 5.1 standard enables the generation of I/Q sample data. While it is known that I/Q signals can be affected by different environmental factors, previous researchers also proposed methods to filter I/Q signals. Yen et al.^[13] developed the filtering algorithm using multiple data packets for angle estimation under an antenna array system (single anchor). They first separated the packets within a given time window into three classes based on the phase difference values. Then, they developed a PDDC to choose a dominant cluster from the class with the largest number of packets. The final angle is estimated from the mean value calculated with all the phase differences within the cluster. For all the I/Q samples within one data packet, Hajiakhondi et al.^[14] developed the nonlinear least squares curve fitting on the raw data to reduce the noise effect. Then, the Kalman filter was applied to the phase difference to address the phase shift problem of devices. Finally, considering the various interference among different BLE channels, they used a Gaussian filter to reduce the difference between ground truth angles and estimated angles to derive the compensation value for each channel. He et al.^[12] also used the nonlinear least square method to filter noise and a Kalman filter to reduce the antenna switching error.

The limitation of previous I/Q sample filtering methods for fingerprint-based IPS is that they are based on limited data packets within a certain time window without global (prior) knowledge. Hajiakhondi et al.^[14] suggested learning the angle compensation vectors for each channel. However, due to the complex and dynamic indoor environment, the fixed compensation value can be greatly affected and become invalid in certain locations and periods. In contrast to the previous methods, this article proposes an attentional filtering layer integrated into the IPS model to infer high-quality packets and learn prior knowledge for filtering noise. As demonstrated in the experiment results, the proposed method effectively removes the noise with the data collected in fewer time steps compared with the traditional filtering methods. It is thus more suitable for real-time positioning.

2.2. IPS with RF Fingerprints

This subsection reviews the related studies that use RF fingerprint data for indoor positioning. We mainly focus on Wi-Fi and BLE data types, which are predominantly adopted in indoor positioning applications.

Wi-Fi-based fingerprints are mainly adopted for indoor positioning in previous research. Microsoft Research proposed the first Wi-Fi fingerprinting system named RADAR,^[19] which computed a user's position with the k -nearest neighbor algorithm and collected the RSS data from the AP side. Recently, researchers and practitioners have gradually adopted CSI as it can provide more prosperous and stable information. The high-dimensional features extracted from the CSI subcarriers can be used as multivariate time series data and fed into machine learning algorithms for classification tasks.^[20] Guo et al.^[21] developed a Wi-Fi-based positioning system that fuses multiple fingerprints gleaned from RSS with multiple classifiers. Experiment results show that their method can combine multiple information well and perform better than previous approaches. Adege et al.^[2] used a multilayer perceptron to train Wi-Fi features consisting of the RSS and basic service set identifier values that were measured from each AP. Wang et al.^[6] presented a deep belief network for Wi-Fi-based IPS with the CSI data. The weights of the proposed model were trained layer by layer using a greedy learning algorithm.^[22] An indoor localization algorithm named MHSA-EC is proposed which is used for solving the problem of effective aggregation of long-distance CSI features and mismatches of long-distance points. The proposed algorithm combines the multihead self-attention mechanism to improve the feature extraction ability.^[23] They proposed Hi-Loc, a hybrid indoor localization system utilizing CSI from the 5G NR network's synchronization signal block. Hi-Loc includes a feature enhancement module, data construction module, and a dual-attention mechanism deep network combining CNN and bidirectional long short-term memory (LSTM). Other studies^[22,23] apply the attention mechanism after data embedding to identify critical feature dimensions (channels) for final prediction. Compared to them, our approach focuses on using the attention mechanism to filter out low-quality data packets before feature embedding. Another study^[24] proposed an IPS model that incorporates the self-attention mechanism to filter the RSS, time of arrival, and angle information of the positioning tag. However, the self-attention mechanism assigns weights to each feature based on its similarity with other input features. BLE packets from different anchors, advertising channels, and directions (including reflected packets) can generate highly diverse features. In the real-time positioning process, the limited number of packets (especially when reflected packets dominate) for each time of prediction can result in large variations in features, significantly impacting the weight assignment and the final positioning performance.

For BLE fingerprint-based IPSs, Faragher and Harle^[4] used the BLE fingerprints obtained from 19 beacons distributed in a 600 square-meter environment for positioning and investigating key parameters, including beacon density, transmit power, and transmit frequency. Their experiment results show higher accuracy than the IPS based on Wi-Fi-based networks. Ishida et al.^[25] proposed a BLE-based fingerprinting localization scheme. They measured RSS on each advertising channel with channel mask operations and sent the advertising packets over a specific duration. The fingerprints of each location consist of the time-average RSS from all BLE beacons. They used one norm distance to evaluate the calculated fingerprint and reference fingerprints. As there is a lack of open BLE datasets, Baronti

et al.^[26] proposed an indoor BLE dataset and explored some used cases such as localization, tracking, occupancy, and social interaction. The above IPSs rely on BLE RSS values which are susceptible to interference and contain limited information. They affect the practical application of these methods.

Further, researchers also developed BLE 5.1 fingerprint-based localization systems. Hajiakhondi et al.^[11] proposed a CNN-based AoA model for IPS with BLE 5.1. The raw AoA measurement data are converted to spatial spectra with the noise eigenvectors and reshaped into 2D matrices. Then the constructed CNN model extracts the latent features from the matrices to predict the ground truth coordinates of each tag. Babakhani et al.^[10] investigated AoA estimation by combining the spatial power spectrum information derived from PDDA and recurrent neural network. Koutris et al.^[9] developed multiple DL model structures for AoA estimation and then used the least squares algorithm to estimate the tag's position. These studies, however, do not well address the data interference problem, which is inevitable for practical IPS.

3. Proposed Deep Learning-Based BLE 5.1 AoA IPS

3.1. Fingerprint-Based BLE 5.1 AoA IPS

The new BLE 5.1 AoA IPS was developed based on the Minew BLE 5.1 AoA G2 system with four anchors as the receivers and the E5 beacon tag as the transmitter, as shown in **Figure 1**. The anchors are installed at various locations within the application environment. Each BLE 5.1 tag makes known its position by periodically broadcasting BLE packets on three advertising channels. The BLE 5.1 anchors then receive these packets. Each anchor has multiple antennas, enabling them to sample the I/Q data from CTE (the additional field of BLE 5.1 packets following the cyclic redundancy check^[8]) through antenna switching. The collected I/Q sample data from all anchors are then transmitted to a central gateway. They are then used to generate the input features for the proposed DL model AnFIPNet for estimating the position of the tag.

Next, let us explain how we construct the input features from the measured I/Q sample data. The feature construction process is illustrated in **Figure 2**. As mentioned in Section 1, one of the difficulties of using the I/Q sample data from multiple anchors is

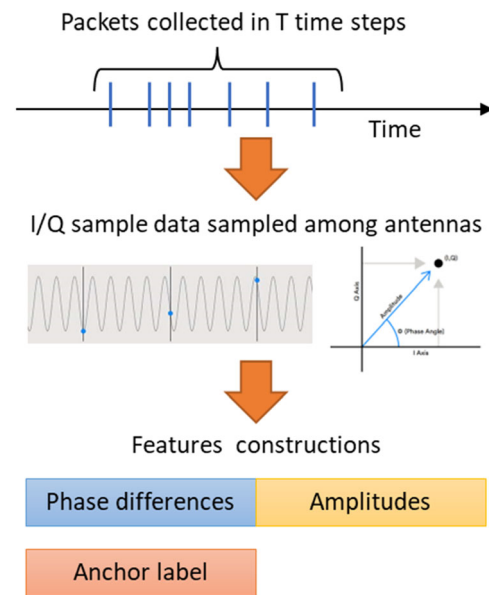


Figure 2. The feature construction process of the proposed multianchor BLE 5.1 IPS.

the irregular arrival time of the data packets. To solve the problem, we construct the input features based on the position information of the BLE packets received from the anchors at a fixed number of “time steps” rather than a fixed time interval as in other approaches. Specifically, within a time period, a BLE 5.1 system can collect I/Q data sampled from a series of BLE packets received by multiple anchors. Each received BLE packet provides the position information of a tag at a time step, which does not characterize the actual time but rather an event. As shown in Figure 2, there is a varying amount of time between two time steps. It is to ensure that a sufficient amount of data is received for feature construction. We assume that anchors can generate N I/Q sample pairs at each time step $t \in \{1, \dots, T\}$ (i.e., when a BLE packet is received). Then, the raw I/Q sample data collected from T time steps is given by $R = \{P_1, P_2, \dots, P_T\}$, where $P_t = \{I_{t,1}, Q_{t,1}, \dots, I_{t,N}, Q_{t,N}\}$. In this way, a regular data flow is generated to the subsequent deep neural network for position estimation.

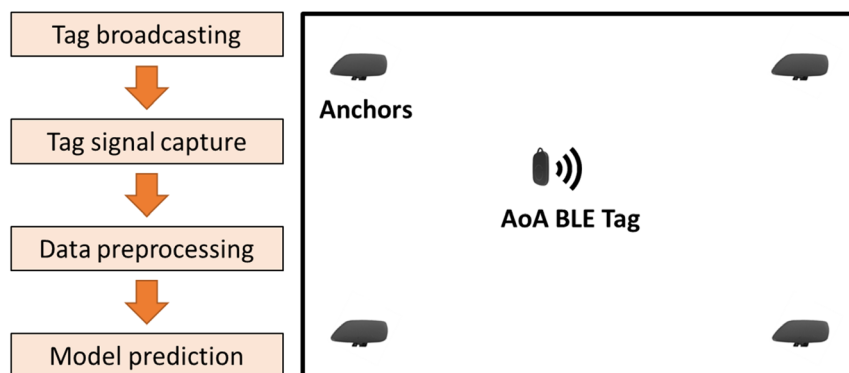


Figure 1. IPS with BLE 5.1.

While the phase differences of I/Q samples can well capture the signal direction information and are widely adopted by previous research,^[9,13] the correlations of the amplitude (signal gain) of I/Q data sampled by each antenna can also provide useful information to construct unique features. Based on these considerations, we adopt the amplitude and phase difference of I/Q sample data to construct the features for each location. For the amplitude feature, we calculate the amplitude of each I/Q sample and put them into a vector with N length. Then we generate the phase difference feature. For the I/Q samples derived in the reference period, we directly use their phase. For the I/Q samples derived in the switch/sample slots, under the predefined switch pattern, we calculate the phase difference between each I/Q sample and one predefined I/Q sample of the reference period. Finally, we normalize the two features and concatenate them into a $2N$ length vector.

Following the above approach, the I/Q sample data collected in T time steps is structured as a $T \times 2N$ feature matrix, which is denoted by $X \in \mathbb{R}^{T \times 2N}$. Each row of X corresponds to a BLE packet received by anchor a , where $a \in 1, \dots, A$. Here, A denotes the total number of deployed anchors. Then, the anchor label for the constructed feature matrix X is denoted by $x_c = \{x_{ac,1}, x_{ac,2}, \dots, x_{ac,T}\} \in \mathbb{R}^T$, where $x_{ac,t}$ represents the corresponding anchor index for the t -th received BLE packet. The ground truth position label for each location is defined by $Y_D \in \mathbb{R}^D$, where $D = 1, 2, \text{ or } 3$ refers to the number of dimensions of the position label.

In summary, we mainly focus on two points to improve the integration of BLE 5.1-based I/Q sample data with DL-based positioning models. First, we utilize time steps rather than fixed time intervals to control better the number of BLE packets used in constructing input data. This helps prevent positioning instability caused by the packet scarcity that can hinder the effectiveness of each location estimation. Second, we combined amplitudes and phase differences as features to fully exploit the feature extraction capabilities of DL models.

Then, the remaining research problem can be defined as follows. Given an input feature matrix X and corresponding anchor label x_{ac} extracted from the I/Q sample data in T time steps, estimate $\hat{Y}_D \in \mathbb{R}^D$ for each tag location.

Remark 1

The gateway of the BLE positioning system generally collects one processed I/Q sample data from one anchor at a time and is set with a fixed upload interval. Due to various environmental and hardware interferences, each anchor shows up randomly in the anchor label x_{ac} of T time steps. Besides, the interference affects the transmission and reception of Bluetooth signals, and the gateway cannot perform a 100% scan which leads to packet loss. The server may receive I/Q sample data after multiple upload intervals. This is why we do not require a fixed time interval between two continuous time steps.

Remark 2

To streamline the dataset construction and model training, we do not specify the broadcasting channel information in the

constructed data feature. As the system often needs a longer period to ensure that at least one packet is collected from each channel and anchor, merging the data from three channels allows us to use data of fewer time steps to build one model input. Fewer time steps also enhance the real-time positioning capability (use the data collected in a shorter time period) in the actual positioning process.

Remark 3

Our data structure is different from the traditional multiple anchors/frequency data and multiple general instances learning data. The multiple anchors/frequency data adopted in the previous research^[10,11] can be considered as one kind of multivariate time series data,^[27] in which each sensor has exactly one data feature. Based on the anchor information, our model input data can have multiple features from one anchor (sensor). The “redundant” data are effectively utilized to improve the prediction performance. Besides, compared with the multiple instances learning classification,^[28] our data has additional sublabel information (anchor), which can be utilized for network learning and feature extraction.

3.2. Proposed Indoor Positioning Model AnFIPNet

In this section, we first present the overall architecture of the proposed deep neural network model AnFIPNet for BLE 5.1 fingerprint-based positioning. Then, the model functional blocks are explained in detail in the following subsections.

3.2.1. Model Architecture Overview

Figure 3 illustrates the overall architecture of AnFIPNet, which comprises two main subnetworks: attentional filtering network and position estimation network. The input of the model is the feature matrix X detailed in Section 3.1 and is constructed from the BLE packets received in T time steps. As shown in **Figure 3**, all features from the same anchors are separated into different groups. An attentional filtering block is applied to filter outliers and choose high-quality features from each anchor. They are then sent to the position estimation network where a feature embedding block is developed to learn embedded representations. After the feature embedding, we implement an average pooling on the anchor dimension and derive the fusion representation for the final estimation. The model’s output is the predicted location of the tag. The attentional filtering and feature embedding blocks are shown in detail in **Figure 4** and illustrated next.

3.2.2. Attentional Filtering Block

With abundant training data, we propose an attentional filtering block integrated into AnFIPNet. The design is conceived under the consideration that traditional methods filter the noise in the data by performing a weighted sum of the data point directly or iteratively. The weights are determined manually or through an optimization process. We follow this weighted sum approach but learn an attentional filtering block, as shown in **Figure 4a**, to

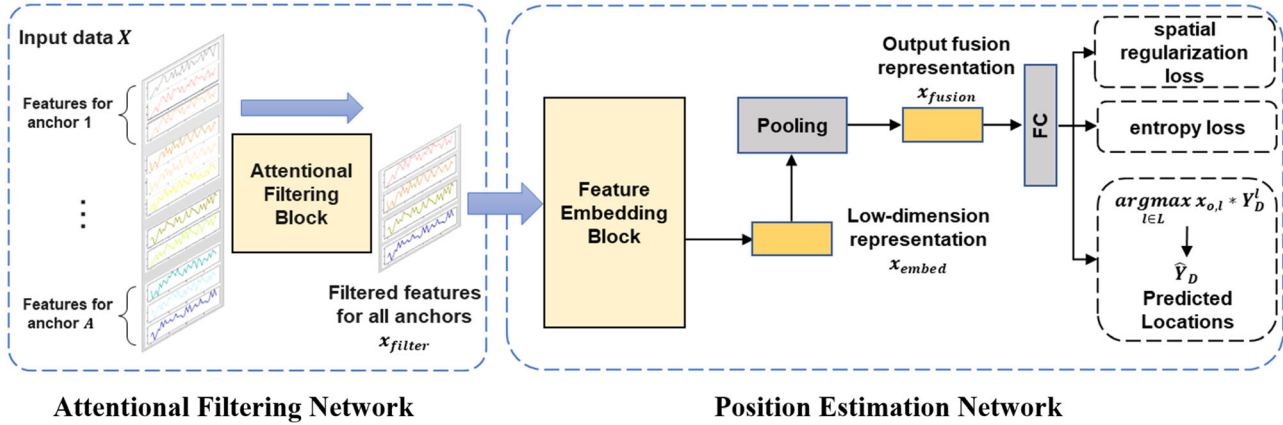


Figure 3. Overall architecture of AnFIPNet.

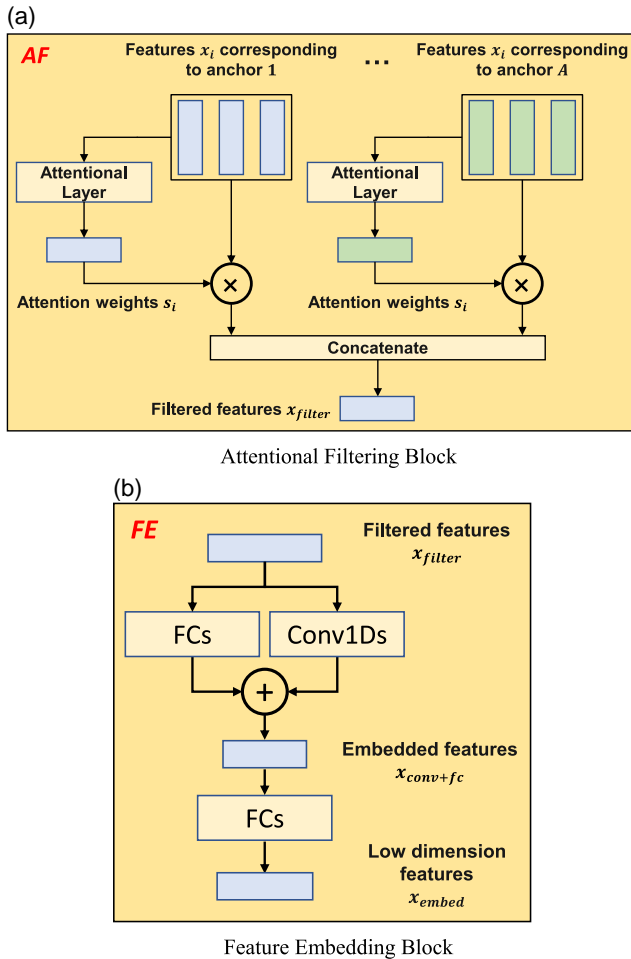


Figure 4. Detailed structures of blocks of AnFIPNet. a) Attentional Filtering Block, b) Feature Embedding Block.

generate the weights dynamically based on the quality of the input feature. Smaller weights will be applied to the features that are predicted to be of lower quality by the attentional filtering block. They are thus suppressed when combined with other

high-quality features. Note that the signals received by different anchors can exhibit different erroneous behaviors.^[8] Therefore, we train a separate filtering layer for each anchor. The filtering layer dynamically assigns weights to the features of each anchor. Let x_i represent the i -th constructed feature (the i -th row) in X . For each x_i , the trainable attention weights s_i can be computed by^[29]

$$s_i = w_a^T \tanh(V_a x_i^T), i \in [1, T] \quad (1)$$

where a is the corresponding anchor index of the feature x_i ($a = x_{ac,i}$). The superscript T refers to matrix transpose. $w_a \in \mathbb{R}^{d_f \times 1}$ and $V_a \in \mathbb{R}^{d_f \times 2N}$ are the trainable parameters of the filtering layer for anchor a . d_f is the size of the hidden dimension. The attentional filtering block can also stack multiple filtering layers to obtain the final weights. Then, for all features, the weights are normalized further via a SoftMax operation as follows.

$$s_i = \frac{\exp(s_i)}{\sum_{x_{ac,j}=x_{ac,i}} \exp(s_j)}, i \in [1, T] \quad (2)$$

The sum of the attention weights is equal to 1. Finally, the features of anchor a are filtered by the following normalized weighted sum operation.

$$x_a = \sum_{x_{ac,i}=a} x_i \times s_i \quad (3)$$

An input feature X may comprise different numbers of features from each anchor. To enable batch handling of feature matrices, we propose a padding-and-mask operation to allow the features of each anchor to have the same shape. First, we separate the input data X into A clusters according to the corresponding anchor label x_{ac} . Next, from the dataset, we derive the maximum number of packets G collected from one anchor within T time steps. Then, for each anchor cluster, we pad the corresponding features with zeros to ensure it has G rows of data. Finally, we transform the input X from the structure $\mathbb{R}^{T \times 2N}$ to $\mathbb{R}^{A \times G \times 2N}$. For example, assume that an input feature X contains the data of ten time steps (ten packets) collected from four anchors and the

feature dimension is 100 ($2N$). In the training dataset, it is determined that a feature should have a maximum of eight packets from one anchor; thus, $G = 8$. Assume that X has three packets received from anchor 1. We extract these packets and pad them with zeros to form an input feature with a shape of 8×100 . We do the same operations for the features from other anchors. Finally, X is transformed from $\mathbb{R}^{10 \times 100}$ to $\mathbb{R}^{4 \times 8 \times 100}$.

When padding the data for each cluster, we simultaneously generate a binary mask $M \in \{0, 1\}^{A \times G}$, where a '1' indicates a zero-padded row. Then, the model takes the padded feature matrix X and the corresponding mask M as input pairs. The features and mask for anchor a are denoted by $x_a \in \mathbb{R}^{G \times 2N}$ and $m_a \in \{0, 1\}^G$. In the attentional filtering layer, the weight for the original data dimension and padded dimension will be calculated together. When normalizing the weight, we only consider the dimension of the anchor data. Thus, the SoftMax operation is applied to the original data dimension and implemented as follows.

$$A_a = \text{softmax}([\mathbf{w}_a^T \tanh(V_a x_a^T)] \odot (1 - m_a)) \quad (4)$$

where the symbol \odot stands for elementwise multiplication. Finally, after filtering, we concatenate the features of all anchors and derive the output $x_{\text{filter}} \in \mathbb{R}^{A \times 2N}$.

According to the SoftMax formulation, SoftMax can substantially decrease the attention assigned to low-quality features. Nevertheless, it remains unable to reduce these attention weights to zero, and certain packets can still impact the outcome of fused features and following embedding processes. To solve this problem, we use attention thresholding after Equation (4).^[30]

$$A_{a,g} = \begin{cases} A_{a,g} & \text{if } A_a > \gamma \\ 0 & \text{else} \end{cases}, \quad g \in 1, \dots, G; a \in 1, \dots, A \quad (5)$$

where γ is the threshold value, and g is the index of packets.

3.2.3. Feature Embedding Block

In the feature embedding part, as shown in Figure 4b, we apply both the CNN and FC layers to learn the correlations of the antennas and extract embeddings from each anchor's features. CNN generally pays more attention to local features, while the FC layer can keep the global receptive field without losing sample correlation information.

We use separate parameters for the amplitude and phase difference features for all layers to facilitate the learning process. Therefore, we split the filtering output feature x_{filter} into amplitude and phase differences and concatenate them in the anchor dimension as $\tilde{x}_{\text{filter}} \in \mathbb{R}^{2A \times N}$. Then, we input $\tilde{x}_{\text{filter}}$ to both embedding layers. The CNN consists of three 1D convolutional layers. Each convolutional layer is followed by batch normalization and leaky rectified linear unit (leaky ReLU).^[20,31] The number of filters for all convolutional layers is set to $2A$ (individually for each feature). We set the kernels of the three convolutional layers to have the sizes of 8, 5, and 3. The output of the CNN is $x_{\text{conv}} \in \mathbb{R}^{2A \times d_c}$, where d_c is the output dimension of the final convolutional layer. Similarly, the output of the FC layer is represented as $x_{\text{fc}} \in \mathbb{R}^{2A \times d_f}$, where d_f is the corresponding output dimension. Then we concatenate the outputs together and get $x_{\text{conv+fc}} \in \mathbb{R}^{2A \times (d_c + d_f)}$. Finally, we use an FC layer to map $x_{\text{conv+fc}}$ to the low-dimension feature $x_{\text{embed}} \in \mathbb{R}^{2A \times d_e}$, where d_e is the dimension of the final embedding.

3.2.4. Location Estimation and Loss Function

After generating the feature embeddings, we apply an average pooling on the $2A$ dimensions and derive the output fusion representation.

$$x_{\text{fusion}} = \text{pooling}(x_{\text{embed}}) \quad (6)$$

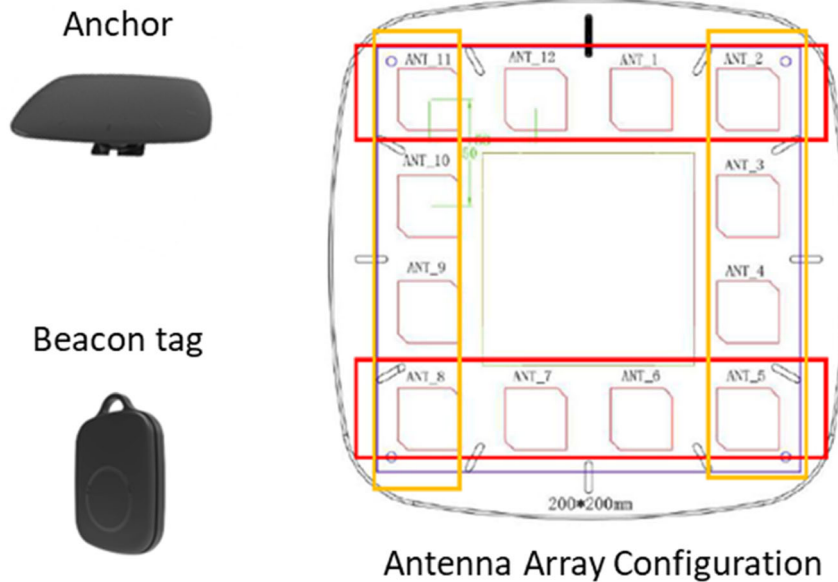


Figure 5. BLE 5.1 IPS devices.

where $x_{\text{fusion}} \in \mathbb{R}^d$. Finally, the fusion representation is fed into an FC layer and produces the output $x_o \in \mathbb{R}^L$, where L is the number of locations in an indoor environment. x_o gives the probabilities of the current tag at all grid points. We choose to train the model with the entropy loss, which achieves better performance than the regression loss. The entropy loss is also adopted by the previous fingerprint-based IPS models.^[16] Then, the IPS model is trained via backpropagation by minimizing the entropy loss.

$$\text{Loss} = -\frac{1}{M} \sum_{m=1}^M Y_D^m \times \log(\hat{x}_o^m) \quad (7)$$

where m is the index of the elements in a minibatch, Y_D^m is the corresponding one-hot vector of the ground truth label, and \hat{x}_o^m is the normalized one-hot vector estimated location through the SoftMax function. M is the size of a minibatch. Furthermore, to facilitate the evaluation with different performance metrics, we generate an estimate of the final location through the final layer output.

$$\hat{Y}_D = (\text{argmax}_{o \in L}(\hat{l})) \times Y_D(l) \quad (8)$$

Even though entropy loss often leads to better training performance than regression loss, it fails to consider the spatial information of individual grid points, which may result in sub-optimal feature learning. To further improve performance under the entropy loss training, we propose incorporating spatial relationships between the grid points. It leverages the coordinate information of each point to increase the discrimination among embeddings. Let S_l represent the set of input data belonging to grid point l and E_l represent the corresponding value vector derived from the average of fusion representations.

$$E_l = -\frac{1}{|S_l|} \sum_{y^m=l} x_{\text{fusion}}^m \quad (9)$$

Based on the average fusion representations, we propose the spatial regularization loss \mathcal{L}_{sr} with the radial basis kernel function and distance among grid points

$$\mathcal{L}_{\text{sr}} = \frac{1}{L^2} \sum_{k_1, k_2 \in [1, L]} D_{k_1, k_2} e^{-\frac{\|E_{k_1} - E_{k_2}\|_2}{\sigma}} \quad (10)$$

where D_{k_1, k_2} is the norm-2 distance between the grid point k_1 and k_2 . σ is a scale factor and can be set to 1. For \mathcal{L}_{sr} , we assign the loss weights based on distances, meaning that grid points located far from each other will have greater differences between their average fusion representations. Finally, our training loss function becomes

$$\text{Loss} = -\frac{1}{M} \sum_{m=1}^M Y_D^m \times \log(\hat{x}_o^m) + \mathcal{L}_{\text{sr}} \quad (11)$$

4. Experimental Section

In this section, we first describe the experiment setup including the details on the dataset's construction, performance metrics,

and baseline methods. Then we evaluate the performance of the proposed IPS model compared with the baseline approaches. Finally, we present an ablation analysis to illustrate the effectiveness of the proposed modules. All computations were conducted on DGX A100 GPUs with 40 GB memory and AMD EPYC 7742 64-Core Processor.

4.1. Experimental Setup

4.1.1. Dataset Construction

As mentioned in Section 3.1, the developed IPS is based on the Minew BLE 5.1 AoA G2 system with four anchors as receivers and the E5 Beacons tag as the transmitter (**Figure 5**). Using the I/Q sample data obtained from the system, we construct the data features for feeding to the proposed AnFIPNet. Note that although the proposed IPS is built on the Minew G2 system, it can be easily extended to other BLE 5.1 hardware platforms. For the Minew G2 system, each anchor consists of 12 antennas, arranged into four linear antenna arrays perpendicular to each other. The tag broadcasts BLE packets on three broadcasting channels. The broadcasting interval was set to 100 ms. The upload interval of the gateway was set to 100 ms. Under such a setting, our system can receive about five packets per second. By decreasing the broadcasting interval to 50 or 20 ms, we can have higher packet receiving rates.

Based on the Minew G2 system and the proposed feature construction method, we created two datasets with 2D coordinates in the laboratory and office scenarios to train and evaluate the proposed AnFIPNet. The datasets will be released for public download. The laboratory scenario was set up on the first floor of Building 16 W at the Hong Kong Science Park. It is a rectangular area separated by cubicles, as shown in **Figure 6a**. The total area covered is about 10 m² with a horizontal span of 4 m and a maximum vertical span of 2.5 m. We laid down a grid of 24 (6 × 4) points with 50 cm grid spacing covering the walking surface of the area. We measured the coordinates of each grid point in cm following a coordinate system starting at the bottom left in **Figure 6a**. Anchors were placed at the corners, facing the ground at 1.83 m high. We collected around 2 min of data at each grid point, and four different orientations (East, South, West, and North) were considered, as in the previous approach.^[26] The BLE tag was put on the ground, facing each of the orientations for 30 s. Within the data collection time, four anchors collect BLE data simultaneously and all data are sent to the server through the gateway. All collected samples were used for model training and testing. The office scenario was set up on the 12th floor of Building 19 W at the Hong Kong Science Park, as shown in **Figure 6b**. The total area covered is about 24 m² with a maximum horizontal span of 6 m and a maximum vertical span of 4 m. We laid down a grid of 55 points with 50 cm grid spacing. Anchors were placed at the corners, facing the ground at 1.7 m high. We also collected 2 min of data at each grid point, and four different orientations of the tag were considered. The BLE tag was placed on a tripod 1 m high, facing each of the orientations for 30 s. In both scenarios, the anchors sent the extracted I/Q sample data to a gateway. Finally, the gateway uploaded the collected data to a server for storage. Note that since the data are measured with a real-world BLE 5.1 system, measurement errors are inherent in the samples

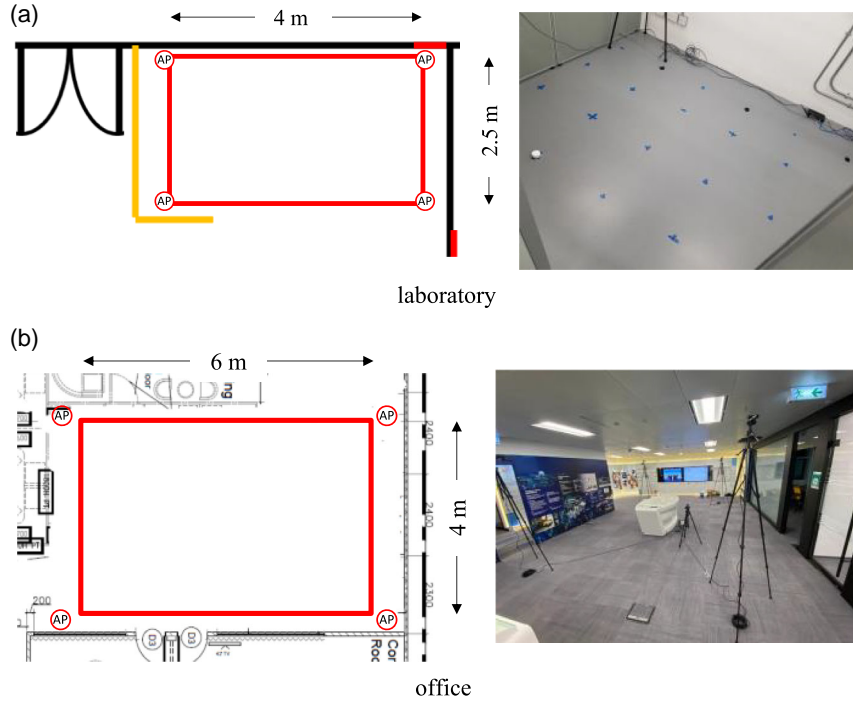


Figure 6. Datasets collection environments. The anchor (AN) locations are circled. a) Laboratory, b) office.

obtained. They lead to the estimation errors made by different positioning approaches, as shown in the experiment results.

As mentioned above, each input feature is constructed by several BLE packets. Each BLE packet is constructed by the 34 I/Q samples obtained in an antenna switch period among the 12 antennas of the anchor plus the 4 I/Q samples obtained from the reference period (total 38). When constructing the two datasets, the packets received from each grid point were grouped in time steps $T = 10$ and 20 to form the input features (in each time step, a BLE packet was received). Thus, two sets of input features of different time steps were obtained for each dataset. We choose the minimum T as 10 to ensure that most of the input features (95%) have at least one BLE packet received by each anchor. In the online prediction process, with the receiving rate of 5 packets per second, using 10 or 20 BLE data packets will require 2–4 s to obtain one predicted tag's location. Note that the tag broadcasting interval can be set shorter (50, 20 ms). In this case, less time is required to get a positioning result. Therefore, the BLE 5.1-based IPS can be implemented for real-time localization. With $T = 10$ (20), the 16 W lab dataset contains 1509 (740) input features, and the 19 W office dataset contains 3211 (1592) input features. After separation, the number of input data for each grid point becomes less than the number of grid points (classes). To ensure the comparison quality, we used 80% of the input features for training and 20% for testing. This segmentation was also adopted by previous fingerprint-based IPS studies.^[32]

4.1.2. Performance Metrics

To evaluate the positioning performance of AnFIPNet and other methods from previous research on BLE 5.1 IPS, we use two widely adopted metrics, that is mean square error (MSE).

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \quad (12)$$

and mean distance error (MDE).

$$\text{MDE} = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \quad (13)$$

where (\hat{x}_i, \hat{y}_i) and (x_i, y_i) are the estimated and ground truth coordinates, respectively; N is the total measurement number. Moreover, we utilize the standard deviation of distance error (SDDE) to measure the stability of positioning results.

$$\text{SDDE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} - \text{MDE})^2} \quad (14)$$

4.1.3. Baseline Methods

We compare the proposed AnFIPNet with the following methods: (MUSIC) algorithm;^[33] PDDA;^[18] MHSA-EC;^[22] Hi-Loc;^[23] attention-based IPS model by Tang et al.;^[24] Gaussian–Bernoulli restricted Boltzmann machine plus liquid-state machine (GBRBM + LSM);^[34] time series attentional prototype network (TapNet);^[20] CNN-based joint APs model by Koutris et al.;^[9] and 2D image CNN by Hajiakhondi et al.^[11] For the MUSIC and PDPA algorithm, due to various interference and complex indoor environments, we remove 10% maximum and minimum phase difference outliers of each anchor to mitigate the phase difference fluctuations. After deriving the

angle estimations from anchors, we use the least squares to estimate the final 2D positions. All methods are implemented in Python.

To implement the models MHSA-EC,^[22] Hi-Loc,^[23] GBRBM + LSM,^[34] TapNet,^[20] Hajiakhondi, et al.,^[11] and Koutris, et al.^[9] on our datasets, we transform the data into the default format defined in their original papers. The data construction process is illustrated as follows. For the I/Q samples collected in T time steps, with the anchor information, we choose one feature from each anchor and concatenate them into a matrix. We choose the final one when there are multiple data points. If there is no data for an anchor, we set all the values in the corresponding row of the input feature to zero. Other configurations for models MHSA-EC,^[22] Hi-Loc,^[23] GBRBM + LSM,^[34] TapNet,^[20] Hajiakhondi, et al.,^[11] and Koutris, et al.^[9] are according to their basic settings. We train our model using the Adam optimizer with an initial learning rate of 0.0001 and a weight decay rate of 0.001. We use a single filtering layer with 128 hidden units for each anchor. The attention threshold is set to 0.01.

The corresponding time complexity and model size comparison are shown in **Table 1**. The time complexity of the MUSIC and PDDA algorithms depends on the number of sensors K , the number of timesteps T , the number of anchors A , and the scanning step angle δ .^[9] In our scenario, A is set to 4. The BLE anchor consists of four antennas arranged in a row, yielding $K = 4$, and the scanning step angle δ is set to 1° . The time complexity of the GBRBM + LSM^[34] depends on the number of anchors A , the feature dimension N , the number of visible units N_v and the number of hidden units N_h . The feature dimension N is 76 (amplitude and phase difference of I/Q sample data). The visible and hidden units are set to 304 and 90, respectively. The time complexity of Koutris^[9] and Hajiakhondi^[11] arises from the convolutional layers. In the study by Koutris,^[9] the input data and channel dimension correspond to the number of anchors A , the feature dimension N , and the frequency dimension F . BLE 5.1 uses three advertising channels, and F is set to 3. In Hajiakhondi,^[11] the data is resized to 28×28 (N_H and N_W are 28) and one input channel. Tapnet's complexity depends on its CNN layers and the attentional prototype learning module.^[20] The latter involves calculating a weighted sum of distances to the prototype embedding for each class. L represents the

number of locations. Compared to other models, its larger model size stems from establishing embedding modules for each class (location). As the number of classes increases, the model's scale grows. The time complexity of Hi-Loc primarily depends on the bidirectional LSTM (biLSTM) layer, where the hidden state dimension d_h is set to 64.^[23] In the study by Tang et al.^[24] and MHSA-EC,^[22] the time complexity primarily depends on the self-attention modules. Specifically, MHSA-EC includes three attention modules. AnFIPNet's time complexity is influenced by the preceding attentional filtering block and the CNN blocks. G represents the maximum number of packets collected from a single anchor within T time steps, as discussed in Section 3.2.2.

We employ the early-stopping approach for all methods to determine when to stop training. Specifically, we partition 10% of data from the training set as the validation set and define an iteration number of 1000 to wait for the validation loss improvement in both datasets. Training is stopped if the validation loss does not improve within the defined iterations. The training time and execution time for all methods are shown in **Table 2**. The execution time reflects the time required for the model to make predictions. The MUSIC and PDDA algorithms do not require training. Their execution time includes using least squares to estimate the final 2D positions. For the machine learning method GBRBM + LSM, it is trained on the CPU and there are two numbers in each cell. The first number represents the training time for the GBRBM. We observe a continuous loss decrease until reaching the maximum iteration limit (10 000). The second number represents the training time for LSM.

The results show that apart from GBRBM + LSM and Hi-Loc, the training time of most methods can achieve convergence within a few minutes. In all DL methods, Hi-loc consumed the most GPU training time. The slower training in Hi-loc stems from its biLSTM module. For each method, the differences in training time across different datasets and time steps are attributed to variations in dataset sizes. For instance, the 19 W dataset (3211 instances when $T = 10$) is larger than the 16 W dataset (1509 instances when $T = 10$), and the datasets have more instances when T is smaller (e.g., in the 19 W dataset, there are 3211 and 1592 instances when $T = 10$ and 20). Since we employed full-batch training, the computation time per iteration increases with the datasets, resulting in a longer overall convergence time. For

Table 1. Time complexity and model size of different approaches.

Methods	Time complexity	Model size 16 W [MB]	Model size 19 W [MB]
MUSIC ^[33]	$O(AK^2T + AK^3 + AK^2(180/\delta))$	–	–
PDDA ^[18]	$O(AKT + AK(180/\delta))$	–	–
GBRBM + LSM ^[34]	$O(ANN_vN_h)$	0.03	0.03
Koutris ^[9]	$O(ANF)$	0.07	0.07
Hajiakhondi ^[11]	$O(N_HN_W)$	0.05	0.06
Tapnet ^[20]	$O(LNA)$	1.86	3.05
Hi-Loc ^[23]	$O(Td_h^2 + Td_hN)$	0.06	0.06
Tang et al. ^[24]	$O(T^2N)$	0.02	0.02
MHSA-EC ^[22]	$O(A^2N)$	0.07	0.07
AnFIPNet	$O(AGN)$	0.17	0.18

Table 2. Training time and execution time of different approaches. The unit is second (s).

Dataset	T	10		20	
	Methods	Training time	Execution time	Training time	Execution time
16 W	MUSIC ^[33]	–	0.1287	–	0.1689
	PDDA ^[18]	–	0.0566	–	0.1126
	GBRBM + LSM ^[34]	3068.9843 + 650.8704 (CPU)		2761.6728 + 358.91361 (CPU)	
	Koutris ^[9]	28.4096	0.0009	21.1062	0.0008
	Hajiakhondi ^[11]	49.3352	0.0010	16.3836	0.0011
	Tapnet ^[20]	75.1713	0.0047	40.8318	0.0048
	Hi-Loc ^[23]	520.5738	0.0016	258.9775	0.0019
	Tang et al. ^[24]	156.4610	0.0032	69.7960	0.0084
	MHSA-EC ^[22]	153.4828	0.0134	133.9793	0.0135
	AnFIPNet	158.7889	0.0037	126.3775	0.0038
19 W	MUSIC ^[33]	–	0.1304	–	0.1654
	PDDA ^[18]	–	0.0573	–	0.1131
	GBRBM + LSM ^[34]	3686.7621 + 1737.7620		3087.9884 + 1102.1073	
	Koutris ^[9]	56.5130	0.0011	40.6431	0.0009
	Hajiakhondi ^[11]	66.8104	0.0011	36.7363	0.0012
	Tapnet ^[20]	94.1786	0.0063	47.0366	0.0062
	Hi-Loc ^[23]	822.2074	0.0016	566.2631	0.0020
	Tang et al. ^[24]	285.1081	0.0031	112.6071	0.0084
	MHSA-EC ^[22]	452.1883	0.0133	341.4893	0.0133
	AnFIPNet	409.5218	0.0036	298.3035	0.0038

the execution time, it shows that some methods (e.g., the Koutris, Hajiakhondi, and MHSA-EC) are stable across different datasets and time steps according to their time complexity. Some methods exhibited variations in execution time due to their dependence on time steps or class numbers, such as Tang et al. and Tapnet. Most methods are within 0.01 s, which can meet real-time indoor positioning requirements.

4.2. Positioning Performance Comparison

The performance of different methods on the two datasets is shown in Table 3. We observe that the machine learning method GBRBM + LSM achieves a similar performance compared to geometry methods MUSIC and PDDA. All DL methods with fingerprinting data outperform the geometry methods, showing the positioning ability of DL networks with I/Q sample data. We also observe that AnFIPNet gives the best performance and can achieve submeter-level accuracy on both datasets under different numbers of time steps. It is also worth mentioning that other models' performances degrade more on the 19 W office dataset, which has more grid points and the same amount of data (2 min) for each point. For our model, a submeter-level accuracy can still be achieved. It indicates that AnFIPNet can extract more representative features and is robust to larger environments. Besides, AnFIPNet performs better as the number of time steps T increases. It shows that AnFIPNet can extract representative information from the input data containing more packets to improve positioning accuracy. As fingerprinting data collection is time-consuming, in the practical application, the number of

time steps for forming the input features for training and estimation should be well optimized to achieve better positioning/tracking performance. The results also show that different methods exhibit varying levels of SDDE. For instance, in the results using the 19 W dataset with $T = 10$, MHSA-EC^[22] outperforms Tang et al.^[24] in terms of accuracy (0.6544 vs 1.2950 in MDE) but exhibits lower stability (1.1951 vs 0.7490 in SDDE). Notably, AnFIPNet consistently demonstrates the lowest SDDE in all cases, indicating its superior performance in producing reliable and stable localization results.

For the case of the 19 W dataset with $T = 10$, we computed distance error (m) for each test instance and plotted them as a cumulative distribution function in Figure 7. It shows that AnFIPNet can accurately predict the majority of test instances to the true grid points and presents the best accuracy of all methods. Moreover, only a small portion of the data exhibits large distance error, indicating the robustness of the proposed method.

4.3. Ablation Study

4.3.1. Effectiveness of the Designed Feature

To evaluate the proposed training features, we further test the performance of the following variants of AnFIPNet: “AnFIPNet-a”, where the model is trained with only the amplitude features, and “AnFIPNet-p”, where the model is trained with only the conventional phase differences features. The results are shown in Table 4. It can be observed that

Table 3. Performance comparison of different approaches for positioning on two datasets. The units for MSE, MDE, and SDDE are meter square (m^2), meter (m), and meter (m), respectively. Bold formatting represents the best results achieved among various algorithms/variants, and this also applies to the subsequent tables.

Dataset	Methods	10			20		
		MSE	MDE	SDDE	MSE	MDE	SDDE
16 W	MUSIC ^[33]	4.7961	1.4075	1.6778	3.5000	1.2394	1.4014
	PDDA ^[18]	1.4511	1.0750	0.5436	1.3297	1.0291	0.5203
	GBRBM + LSM ^[34]	2.1357	1.2980	0.6716	3.1387	1.6058	0.7485
	Koutris ^[9]	1.0946	0.9686	0.3955	1.0374	0.9443	0.3817
	Hajiakhondi ^[11]	0.4079	0.3555	0.5306	0.2317	0.2117	0.4323
	Tapnet ^[20]	0.0921	0.0614	0.2972	0.3010	0.2111	0.5064
	Hi-Loc ^[23]	0.4093	0.5366	0.3485	0.3757	0.5088	0.3417
	Tang et al. ^[24]	0.2015	0.3657	0.2603	0.1973	0.3607	0.2592
	MHSA-EC ^[22]	0.1614	0.1301	0.3801	0.2225	0.1341	0.4522
AnFIPNet	0.0042	0.0045	0.1361	0.0021	0.0032	0.0805	
19 W	MUSIC ^[33]	35.9914	3.9546	4.5114	23.4158	3.4548	3.3882
	PDDA ^[18]	4.6088	1.9403	1.0198	4.5663	1.8778	0.9186
	GBRBM + LSM ^[34]	6.1779	2.1078	1.3171	6.0334	2.1544	1.1798
	Koutris ^[9]	3.7745	1.8422	0.6173	3.9848	1.8973	0.6206
	Hajiakhondi ^[11]	3.0687	1.3244	1.1466	2.5266	1.1745	1.0711
	Tapnet ^[20]	2.2153	0.8130	1.2467	2.7842	1.0646	1.2848
	Hi-Loc ^[23]	2.0719	1.2340	0.7410	2.0227	1.2047	0.7559
	Tang et al. ^[24]	2.2379	1.2950	0.7490	2.3700	1.3201	0.7921
	MHSA-EC ^[22]	1.8565	0.6544	1.1951	1.1451	0.3893	0.9968
AnFIPNet	0.3240	0.1349	0.5594	0.1793	0.0743	0.2340	

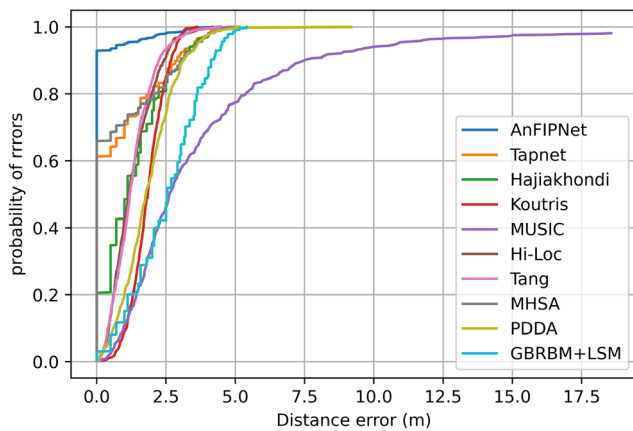


Figure 7. Error probability versus distance error under 19 W dataset with $T = 10$.

AnFIPNet, with the designed feature, achieves the best performance in all metrics. More specifically, the performance of AnFIPNet with the designed feature is better than the performance with only phase differences, by an average of 41.5% MDE reduction in all cases. The results show that useful information can also be captured from the amplitudes to improve the positioning performance. Both the phase and amplitude features are essential for fingerprint-based positioning.

4.3.2. Effectiveness of the Attentional Filtering Block

To show the effectiveness of the proposed attentional filtering block, we further test the performance of the following variants of AnFIPNet with different filtering methods: “AnFIPNet-nf”, “AnFIPNet-af”, and “AnFIPNet-kf”. Under “AnFIPNet-nf” (no filtering), we remove the filtering layer and execute the same feature selection process that is implemented for other fingerprint-based IPS methods. Under “AnFIPNet-af” (average filtering), we use the mean value of the phase difference and amplitude features when there are multiple data points from one anchor. Under “AnFIPNet-kf” (Kalman filtering), we implement Kalman filtering to the features when there are multiple data points from one anchor.^[12,14] The setting for the Kalman filters is not specified in other studies.^[12,14] We adopt the Kalman filtering parameters from another study^[35] for BLE signals in indoor environments. The results are shown in Table 5. It can be seen that the accuracy under three filtering operations is superior to the performance without filtering, indicating that the designed modules for I/Q sample data filtering are essential to BLE 5.1 fingerprint-based IPS. For the SDDE measurement, AnFIPNet exhibits better performance on the 19 W dataset than other variants. Notably, when $T = 20$, AnFIPNet achieves an SDDE of 0.2340, while the best-performing variants only achieve a value of 0.5124. Besides, the AnFIPNet with the proposed attentional filtering gives the best positioning accuracy on both datasets under different numbers of time steps (an average of 53.2%

Table 4. Performances of variants of AnFIPNet trained with different features on two datasets. The units for MSE, MDE, and SDDE are meter square (m^2), meter (m), and meter (m), respectively.

Dataset	T	10			20		
	Methods	MSE	MDE	SDDE	MSE	MDE	SDDE
16 W	AnFIPNet-a	0.0360	0.0276	0.2878	0.0353	0.0160	0.1873
	AnFIPNet-p	0.0070	0.0094	0.1643	0.0065	0.0074	0.1301
	AnFIPNet	0.0042	0.0045	0.1361	0.0021	0.0032	0.0805
19 W	AnFIPNet-a	0.8905	0.3324	0.8832	0.4354	0.1659	0.6387
	AnFIPNet-p	0.5672	0.2201	0.7203	0.2416	0.0909	0.4831
	AnFIPNet	0.3240	0.1349	0.5594	0.1793	0.0743	0.2340

Table 5. Performances of variants of AnFIPNet trained with different filtering operations. The units for MSE, MDE, and SDDE are meter square (m^2), meter (m), and meter (m), respectively.

Dataset	T	10			20		
	Methods	MSE	MDE	SDDE	MSE	MDE	SDDE
16 W	AnFIPNet-nf	0.1086	0.0843	0.4186	0.0083	0.0142	0.1900
	AnFIPNet-kf	0.0223	0.0108	0.1263	0.0047	0.0044	0.0981
	AnFIPNet-af	0.0253	0.0128	0.1372	0.0044	0.0050	0.1228
	AnFIPNet	0.0042	0.0045	0.1361	0.0021	0.0032	0.0805
19 W	AnFIPNet-nf	1.8100	0.7300	1.1301	1.3066	0.4707	1.0417
	AnFIPNet-kf	0.5159	0.2005	0.6397	0.2948	0.1078	0.4322
	AnFIPNet-af	0.5053	0.1914	0.6846	0.2941	0.1029	0.5124
	AnFIPNet	0.3240	0.1349	0.5594	0.1793	0.0743	0.2340

MSE and 37.3% MDE reduction in all cases compared to Kalman filtering). We also observe that as the number of time steps increases (more packets are used to form an input), the performances of average filtering and Kalman filtering on the two datasets also become better. It shows that filtering the data over a longer period can also stabilize the derived feature for model training and inferring. However, using more data to form a feature will increase the time interval of real-time positioning. Our designed attentional filtering layer can achieve better positioning performance with the data collected in fewer time steps. It is thus more efficient in real-time positioning.

4.3.3. Influence of Attention Threshold and Spatial Regularization on Positioning Performance

In this subsection, we evaluate the influence of the attention threshold and spatial regularization on positioning accuracy. We created several variants of AnFIPNet by removing certain operations. These include AnFIPNet_no_reg, which removes the spatial regularization loss \mathcal{L}_{sr} ; AnFIPNet_no_thres, which removes the attention threshold; and AnFIPNet_no_reg_thres, which removes both the spatial regularization loss and attention threshold. The results are presented in **Table 6**.

Table 6. Influence of the attention threshold and spatial regularization. The units for MSE, MDE, and SDDE are meter square (m^2), meter (m), and meter (m), respectively.

Dataset	T	10			20		
	Methods	MSE	MDE	SDDE	MSE	MDE	SDDE
16 W	AnFIPNet_no_reg_thres	0.0162	0.0142	0.1502	0.0140	0.0133	0.1232
	AnFIPNet_no_reg	0.0085	0.0082	0.1396	0.0064	0.0050	0.0861
	AnFIPNet_no_thres	0.0145	0.0151	0.1534	0.0083	0.0076	0.1176
	AnFIPNet	0.0042	0.0045	0.1361	0.0021	0.0032	0.0805
19 W	AnFIPNet_no_reg_thres	0.5302	0.2163	0.7039	0.2606	0.0919	0.3565
	AnFIPNet_no_reg	0.5102	0.1967	0.6867	0.2471	0.0858	0.3117
	AnFIPNet_no_thres	0.3501	0.1519	0.6551	0.1862	0.0772	0.2655
	AnFIPNet	0.3240	0.1349	0.5594	0.1793	0.0743	0.2340

The results show that compared to AnFIPNet_no_reg_thres, both AnFIPNet_no_reg and AnFIPNet_no_thres can improve the metric performances. For example, with the attention threshold, the variant AnFIPNet_no_reg achieves 27.7%, 30.1%, and 13.0% improvements in MSE, MDE, and SDDE, respectively. Then, for all metrics, AnFIPNet outperforms other variants on both datasets. The results suggest that combining the spatial regularization loss with the attention threshold can achieve better positioning performance.

5. Conclusion

In this research, a DL-based multianchor BLE 5.1 indoor positioning system with attentional filtering was developed. The system features a novel deep neural network model named AnFIPNet that effectively estimates the target object's position from its I/Q sample fingerprint. Compared to the existing fingerprint-based positioning methods using BLE 5.1, we introduced an attentional filtering network into AnFIPNet to filter the data collected and extract high-quality features robust to measurement errors of I/Q sample data. It is the first time that a DL approach is used for I/Q sample filtering in IPS applications. For increasing the discrimination among feature embeddings, we also proposed the spatial regularization loss function that provides additional spatial information to the cross entropy-based loss function for training the model. At the system level, we developed a new approach to construct data features from the I/Q sample data. We integrated amplitude features with the phase difference features to allow their correlation to be fully utilized. We also adopted a time-step-based approach for I/Q sample data collection to ensure a steady data flow for the subsequent deep learning operation. To facilitate the research work, we developed two real-world indoor positioning datasets constructed with the developed multianchor BLE 5.1 IPS. They were used in the evaluation of the proposed model. The evaluation results show the following. 1) AnFIPNet achieves superior performance with submeter accuracy on both datasets and significantly outperforms the previous methods. 2) The proposed input features, attentional filtering network, and spatial regularization functions effectively improve positioning accuracy. These results show that the proposed DL-based multianchor BLE 5.1 IPS is an important contribution to the field of study.

Acknowledgements

The work presented in this article was supported by Centre for Advances in Reliability and Safety (CAIRS) admitted under AIR@InnoHK Research Cluster.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are available in the supplementary material of this article.

Keywords

attention-based deep neural networks, BLE 5.1, data filtering, fingerprinting, indoor positioning, interferences

Received: May 31, 2023

Revised: September 28, 2023

Published online: November 16, 2023

- [1] W. Van Woensel, P. C. Roy, S. S. R. Abidi, S. R. Abidi, *Artif. Intell. Med.* **2020**, *108*, 101931.
- [2] A. B. Adege, H.-P. Lin, G. B. Tarekegn, S.-S. Jeng, *Appl. Sci.* **2018**, *8*, 1062.
- [3] S. Monica, G. Ferrari, *Adv. Intell. Syst.* **2021**, *3*, 2000083.
- [4] R. Faragher, R. Harle, *IEEE J. Sel. Areas Commun.* **2015**, *33*, 2418.
- [5] L. Mucchi, P. Marcocci, *IEEE Trans. Wireless Commun.* **2009**, *8*, 1597.
- [6] X. Wang, L. Gao, S. Mao, S. Pandey, *IEEE Trans. Veh. Technol.* **2016**, *66*, 763.
- [7] M. Woolley, *Bluetooth Direction Finding, A Technical Overview*, Bluetooth SIG **2019**, <https://www.bluetooth.com/bluetooth-resources/bluetooth-direction-finding/>.
- [8] M. Cominelli, P. Patras, F. Gringoli, in *Proc. of the 13th Int. Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WINTECH '19)*, Association for Computing Machinery, New York, NY, United States **2019**, pp. 13–20.
- [9] A. Koutris, T. Siozos, Y. Kopsinis, A. Pikrakis, T. Merk, M. Mahlig, S. Papaharalabos, P. Karlsson, *Sensors* **2022**, *22*, 2759.
- [10] P. Babakhani, T. Merk, M. Mahlig, I. Sarris, D. Kalogiros, P. Karlsson, in *2021 Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, Piscataway, NJ **2021**, pp. 1–7.
- [11] Z. HajiAkhondi-Meybodi, M. Salimibeni, A. Mohammadi, K. N. Plataniotis, in *ICASSP 2021–2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Piscataway, NJ **2021**, pp. 7913–7917.
- [12] S. He, H. Long, W. Zhang, in *2021 7th Int. Conf. on Computer and Communications (ICCC)*, IEEE, Piscataway, NJ **2021**, pp. 2160–2164.
- [13] H.-Y. Yen, Z.-T. Tsai, Y.-C. Chen, L.-H. Shen, C.-J. Chiu, K.-T. Feng, in *IEEE 93rd Vehicular Technology Conf. (VTC2021-Spring)*, IEEE, Piscataway, NJ **2021**, pp. 1–5.
- [14] Z. Hajiakhondi-Meybodi, M. Salimibeni, K. N. Plataniotis, A. Mohammadi, in *2020 IEEE 23rd Int. Conf. on Information Fusion (FUSION)*, IEEE, Piscataway, NJ **2020**, pp. 1–6.
- [15] B. Yang, Q. Qiu, Q.-L. Han, F. Yang, *IEEE Trans. Cybern.* **2020**, *52*, 727.
- [16] E. Gönültaş, E. Lei, J. Langerman, H. Huang, C. Studer, *IEEE Trans. Wireless Commun.* **2021**, *21*, 2162.
- [17] F. A. Toasa, L. Tello-Oquendo, C. R. Peñafiel-Ojeda, G. Cuzco, in *2021 IEEE 18th Annual Consumer Communications & Networking Conf. (CCNC)*, IEEE, Piscataway, NJ **2021**, pp. 1–4.
- [18] H. Ye, B. Yang, Z. Long, C. Dai, *IEEE Sens. J.* **2022**, *22*, 7877.
- [19] P. Bahl, V. N. Padmanabhan, in *Proc. IEEE INFOCOM 2000. Conf. on Computer Communications. Nineteenth Annual Joint Conf. of the IEEE Computer and Communications Societies (Cat. No. 0'CH37064)*, Vol. 2, IEEE, Piscataway, NJ **2000**, pp. 775–784.
- [20] X. Zhang, Y. Gao, J. Lin, C.-T. Lu, *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 6845.
- [21] X. Guo, N. R. Elikplim, N. Ansari, L. Li, L. Wang, *IEEE Trans. Ind. Inf.* **2019**, *16*, 3177.
- [22] W. Liu, M. Jia, Z. Deng, C. Qin, *Entropy* **2022**, *24*, 599.
- [23] Y. Ruan, L. Chen, X. Zhou, G. Guo, R. Chen, *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5502415.
- [24] J. Tang, L. Yang, J. Zhao, Y. Qiu, Y. Deng, S. Shen, in *IEEE Int. Conf. on Electronic Technology, Communication and Information (ICETCI)*, IEEE, Piscataway, NJ **2021**, pp. 140–143.

- [25] S. Ishida, Y. Takashima, S. Tagashira, A. Fukuda, in *5th IIAI Int. Congress on Advanced Applied Informatics (IIAI-AAI)*, IEEE, Piscataway, NJ **2016**, pp. 230–233.
- [26] P. Baronti, P. Barsocchi, S. Chessa, F. Mavilia, F. Palumbo, *Sensors* **2018**, *18*, 4462.
- [27] F. Karim, S. Majumdar, H. Darabi, S. Harford, *Neural Networks* **2019**, *116*, 237.
- [28] T. G. Dietterich, R. H. Lathrop, T. Lozano-Pérez, *Artif. Intell.* **1997**, *89*, 31.
- [29] M. Ilse, J. Tomczak, M. Welling, in *Int. Conf. on Machine Learning*, PMLR, **2018**, pp. 2127–2136, <https://proceedings.mlr.press/>.
- [30] C. Jiao, C. Chen, S. Gou, X. Wang, L. Jiao, *IEEE Trans. Cybern.* **2021**, *53*, 124.
- [31] B. Xu, N. Wang, T. Chen, M. Li, Empirical Evaluation of Rectified Activations in Convolutional Network, arXiv preprint arXiv:1505.00853, **2015**.
- [32] F. J. Aranda, F. Parralejo, F. J. Álvarez, J. A. Paredes, *Expert Syst. Appl.* **2022**, *202*, 117095.
- [33] S. Monfared, T.-H. Nguyen, L. Petrillo, P. De Doncker, F. Horlin, in *IEEE 29th Annual Int. Symp. on Personal, Indoor and Mobile Radio Communications*, IEEE, Piscataway, NJ **2018**, pp. 856–859.
- [34] X. Yang, Z. Wu, Q. Zhang, *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1000208.
- [35] U. M. Qureshi, Z. Umair, G. P. Hancke, *Sensors* **2019**, *19*, 3282.