


ORIGINAL RESEARCH

An ensemble approach to deep-learning-based wireless indoor localization

Juthatip Wisanmongkol¹  | Attaphongse Taparugssanagorn¹ | Le Chung Tran² | Anh Tuyen Le³ | Xiaojing Huang³ | Christian Ritz² | Eryk Dutkiewicz³ | Son Lam Phung²

¹Telecommunications Academic Program, ICT Department, School of Engineering and Technology, Asian Institute of Technology, Khlong Nueng, Pathum Thani, Thailand

²School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Wollongong, New South Wales, Australia

³School of Electrical and Data Engineering, University of Technology Sydney, Sydney, New South Wales, Australia

Correspondence

Juthatip Wisanmongkol, Telecommunications Academic Program, ICT Department, School of Engineering and Technology, Asian Institute of Technology, 58 Moo 9, Paholyothin Highway, Khlong Nueng, Pathum Thani 12120, Thailand.
Email: st121010@ait.ac.th

Funding information

New South Wales (NSW) Defense Innovation Network (DIN), Grant/Award Number: DIN Pilot Project Grant (Project ID: 888-006-985)

Abstract

The authors investigate the use of deep learning in wireless indoor localization to address the shortcomings of the existing range-based (e.g. trilateration and triangulation) and range-free (e.g. fingerprinting) localization. Instead of relying on geometric models and hand-picked features, deep learning can automatically extract the relationship between the observed data and the target's location. Nevertheless, a deep neural network (DNN) model providing a satisfactory accuracy might perform differently when it is retrained in the deployment. To mitigate this issue, the authors propose an ensemble method where DNN models obtained from multiple training sessions are combined to locate the target. In the authors' evaluation, several DNN models are trained on the data, which consists of the received signal strength (RSS), angle of arrival (AOA), and channel state information (CSI), used in the existing hybrid RSS/AOA and RSS/CSI fingerprinting, and their root-mean-square error (RMSE) values are compared accordingly. The results show that the proposed method achieves the lower RMSE than the existing methods, and the RMSE can be lowered by up to 1.47 m compared with the ones obtained from a single model. Moreover, for some DNN models, the RMSE values are even lower than the minimum RMSE obtained by their single-model counterparts.

KEYWORDS

indoor communication, indoor navigation, indoor radio, RSSI, learning (artificial intelligence)

1 | INTRODUCTION

Wireless localization has gained an unprecedented interest over the recent years. With an advancement of computing devices and radio frequency (RF) technologies, the need for location-based services has begun to extend beyond military use. Today, wireless localization can be found in a wide variety of industries, including security, commercial, healthcare, and tourism, with applications ranging from intruder detection [1, 2], asset tracking [3–5], and emergency call locating system

[6–8] to capsule endoscope tracking [9–11], geo-fencing [12, 13], and automated museum tour [14–16].

Since first introduced in the 1970s, the global positioning system (GPS) is a well-established radio navigation system for outdoor positioning. As the fundamental operations of the GPS rely on line-of-sight (LOS) communication between the target's receiver and the GPS satellites, the accuracy for indoor positioning is heavily impaired by the obstruction of the LOS path. Consequently, shorter-range wireless technologies, such as Wi-Fi [14, 17, 18], Bluetooth [14, 17, 18], and ultra-wideband

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.

© 2022 Asian Institute of Technology, Thailand. *IET Wireless Sensor Systems* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology.

(UWB) [19–22], have been considered as alternative technologies for indoor and non-line-of-sight (NLOS) localization.

Selecting which type of technology to use depends largely on the requirements of the application at hand. Although the UWB-based localization can achieve an accuracy of a few centimetres, the cost of its infrastructure and the device needed on an object being tracked are much higher than the system based on Wi-Fi or Bluetooth. Conversely, the Wi-Fi-based localization has an accuracy of 1–5 m, but the implementation cost is lower since the infrastructure and related devices are already available in most places. Similar to Wi-Fi, Bluetooth operates in the same 2.4-GHz band and shares an accuracy of 1–5 m. However, its coverage is only around 10–40 m, which makes it impractical for localization in large areas [23]. Therefore, if high-precision localization is the first priority, such as those in capsule endoscope tracking, then choosing UWB is reasonable. For a more relaxed accuracy, such as those in human tracking, then choosing Wi-Fi or Bluetooth is more cost-effective.

The mechanisms used in wireless indoor localization can be broadly categorized into the range-based and range-free approaches [24]. In the range-based approach, such as trilateration and triangulation, the received signal strength (RSS), time of arrival (TOA), and angle of arrival (AOA) are first transformed into their geometric line of positions (LOPs), and the target's location is obtained from the intersection of the LOPs. However, in the range-free approach, these parameters are considered as the fingerprints for the reference locations, and the target's location is obtained by performing fingerprint matching.

In trilateration based on the RSS or TOA measurement, an observed value is transformed into a range information through a RF propagation model. From this range information, a circular LOP having the radius equal to the range is created. To locate the target, an intersection of several circular LOPs has to be determined. This involves solving a set of non-linear equations. To simplify the problem, Ref. [25] proposed a method to linearize the problem so that a simple linear least-squares method (LS) can be used to find the solution. The trade-off for this simplification is, however, the lower accuracy compared with the non-linear LS method. As a way to increase the accuracy, a more complicated subspace method (SS) was proposed in Ref. [25]. By projecting the observed signal into a signal and a noise subspace which are orthogonal to each other, a certain amount of noise is removed from the observation.

More recently, a hybrid RSS/AOA method was proposed in Refs. [26–28]. Based on this method, the same number of the RSS and AOA measurements are required to linearize the problem. However, this puts the burden on the data measurement as an antenna array is required on all measuring devices. To ease this difficulty, Ref. [29] proposed an unbalanced hybrid RSS/AOA method where only one AOA measurement is needed in the data fusion. In Ref. [30], an angle projection method was proposed. By projecting one RSS value onto its x and y components, additional two virtual RSS values are obtained, and methods such as linear LS or SS can be used to locate the target.

Since the accuracy of the range-based approach depends on the quality of the observed signals, the existence of the LOS path is required for accurate positioning. This is, however, not

as crucial in the range-free approach such as fingerprinting. The method is based on the assumption that the observed signals are a unique fingerprint to a specific location, and locating the target is equal to performing fingerprint matching, hence the name. Two phases are required in fingerprinting, namely, offline training and online testing. In the offline phase, the fingerprints for the reference locations are stored in a database called a radio map. In the online phase, the real-time fingerprint is compared with the ones in the radio map for the closest match in order to locate the target.

One of the earliest fingerprinting methods is RADAR [31]. In their work, the RSS measurements are used as the fingerprints, and locating the target is a problem of minimizing the Euclidean distance between the observed fingerprint and the reference ones. However, the RSS tends to fluctuate when there is any change in the environment. Therefore, a probabilistic method called Horus [32] proposed to use the distribution of the RSS as the fingerprint instead of using the RSS values directly. Recent works, such as Refs. [33, 34], are exploring the use of the channel state information (CSI) since it is considered to be more stable and is able to provide finer-grained information compared to the RSS.

Comparing the existing range-based and range-free approaches, the advantages and disadvantages are summarized in Table 1. For large areas with LOS, trilateration and triangulation can be used to reduce the burden of the data collection for the radio map, whereas for smaller areas with many obstructions blocking the LOS path, fingerprinting would be more appropriate. Nevertheless, these methods are still based on predefined geometric models or features that might not be a precise representation of the actual environment. For this reason, the localization accuracy is restrained by the imprecision of these models.

Deep learning is an emerging machine learning tool used in many areas including indoor wireless localization [35–38]. Compared with the existing range-based and range-free localization methods, deep learning does not rely on geometric models or hand-engineered features, which impose a limitation on the localization accuracy. As a universal estimator [39], deep neural networks (DNNs) are able to capture the complexity of the wireless channel through the data-driven learning process, which results in a learnt model that is more robust and resilient to adversarial conditions [40].

In this work, we investigate the use of deep learning in wireless indoor localization to counteract the shortcomings of the existing range-based and range-free localization methods. Several types of DNN models are trained on the same set of data used in the range-based hybrid RSS/AOA and the range-free fingerprinting based on RSS and CSI measurements, and their performance is compared accordingly. For brevity, the two sets of data will be called the range-based and range-free datasets, respectively. For the range-based dataset, the problem is modelled as a regression task, whereas for the range-free dataset, it is modelled as a classification task. With the ability to automatically extract the relationship between the inputs and the outputs, the DNN approach is expected to perform better

TABLE 1 Pros and cons of localization techniques

Range-based approach	Pros: • A radio map is not required prior to real-time positioning Cons: • LOS path is required for accurate positioning • More than one anchor is required
Range-free approach	Pros: • Accurate positioning in NLOS conditions is possible • Positioning using a single anchor is possible Cons: • A radio map is required prior to real-time positioning

Abbreviations: LOS, line-of-sight; NLOS, non-line-of-sight.

than the existing range-based and range-free methods. In most literature, only a single DNN model is considered; however, a model having a satisfactory performance in the development stage might not perform as well when it is retrained in the deployment due to the stochastic nature of the training process. Therefore, we propose a model ensemble method where several models obtained from multiple training sessions are combined to locate the target. By using this method, a more stable target estimation is obtained and localization accuracy could be improved with the help of model diversity. Performance of various types of DNN is explored, including the feed-forward neural network (FNN), 1D-convolutional neural network (1D-CNN), 2D-convolutional neural network (2D-CNN), and long short-term memory network (LSTM), as each of these DNNs operates on the input data differently.

The main contributions of this paper are as follows:

1. Investigation of the performance of the DNN models in indoor localization from both the range-based and range-free perspectives. Although the range-free methods are likely to perform better than the range-based methods in the NLOS situation, and a large amount of literature focusses on applying the DNN to the range-free localization, we investigate the use of the DNN in both approaches, as they have their own merits in different scenarios
2. Investigation of the performance of different types of DNNs, such as FNN, 1D-CNN, 2D-CNN, and LSTM, for indoor wireless localization, and comprehensively compare their performance to the existing range-based hybrid RSS/AOA and range-free RSS/CSI fingerprinting methods
3. Proposal of the model ensemble techniques that combine the outputs from multiple DNN models to achieve a more stable target estimation and higher localization accuracy. The comparison of localization accuracy between a single model and the ensemble model methods are thoroughly presented

The rest of this paper is structured as follows. Section 2 explains the basic concepts of wireless localization and provides the detailed information of the signal parameters used to locate the target. Section 3 summarises the existing range-based localization methods. Section 4 summarises the

existing range-free localization methods. Section 5 provides the descriptions of the proposed DNN models. Section 6 presents the experimentation and the performance evaluation of the proposed DNN models. Lastly, the conclusion is given in Section 7.

2 | BACKGROUND

Wireless localization or wireless positioning is a process of locating an object using radio waves. Generally, a wireless localization system consists of two main entities, namely the target and the anchor. The target is an object in which its location is to be determined, whereas the anchor is a reference object whose exact location is known. By using properties of the propagating waves between these two entities, the location of the target can be estimated.

The computations required for target localization can be carried out at either the anchor or the target. However, when they are power-limited or computation-limited, such as those in wireless sensor networks [41–43], an external location engine can be used to handle these computations. Figure 1 illustrates a wireless localization system consisting of three anchors, a target, and a location engine.

To infer the location of the target, information such as the RSS, TOA, AOA, and CSI measurement is used. In this study, we focus on using the RSS, AOA, and CSI since these parameters can be obtained without using complicated devices. Information on each parameter and how it can be used to localize a target is briefly presented below.

2.1 | Received signal strength

The RSS is a parameter indicating the signal power presented at the receiver. In a NLOS condition, the transmitted signal travels to the receiver through multiple paths. These multipath signals can create either a constructive or destructive effect at the receiver, as described by Equation (1)

$$V = \sum_{p=1}^P |V_p| e^{-j\theta_p}, \quad (1)$$

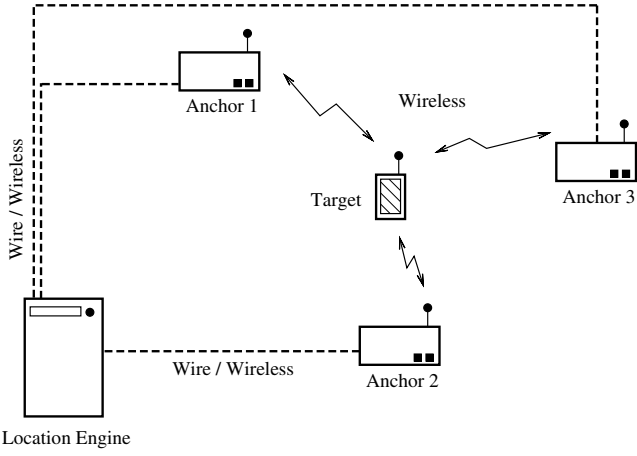


FIGURE 1 A wireless localization system

where V is the overall amplitude, V_p and θ_p are the amplitude and phase of the p th path, and P is the total number of propagating paths. The RSS (dB) is simply the received power, that is,

$$\text{RSS} = 10\log_{10}(|V|^2). \quad (2)$$

As the distance between the anchor and the target increases, the RSS value reduces. Various propagation models can be used to describe the relationship between the RSS and the range information. A common model is the log-normal path loss shown below.

$$\text{RSS}_r = \text{RSS}_{r_0} - 10\gamma\log_{10}\left(\frac{r}{r_0}\right) + w_{\sigma_r}, \quad (3)$$

where RSS_r is the RSS (dB) at distance r from the transmit anchor, RSS_{r_0} is the RSS (dB) at the reference distance r_0 , γ is the path loss exponent, and w_{σ_r} is the shadowing factor that follows the Gaussian distribution with zero mean and standard deviation σ_r . By using the distance or range information obtained from several anchors, the location of the target can be estimated.

2.2 | Angle of arrival

The AOA represents the direction of the incoming signal. In wireless localization, it is used to infer the direction of the target with respect to the anchor. With an assumption that the receiver antenna array is in the far-field region, where the incoming signals can be considered as plane waves, the AOA can be measured with a uniform linear array, as shown in Figure 2. By measuring the time delay of the received signal between the antennas in the array, the observed AOA, θ , can be found from Equation (4) as

$$\theta = \arccos\left(\frac{c\tau}{d}\right) + w_{\sigma_a}, \quad (4)$$

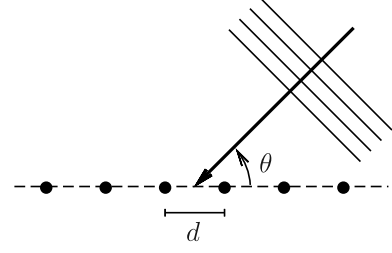


FIGURE 2 Angle-of-arrival measurement using uniform linear array

where c is the speed of the travelling wave, τ is the time delay between adjacent antennas, d is the antenna spacing, and w_{σ_a} is the measurement noise that follows the Gaussian distribution with zero mean and standard deviation σ_a .

2.3 | Channel state information

The CSI has received increasing attention in the field of wireless localization over the recent years. Compared to the RSS, the CSI provides finer-grained information, including the information about each of the multipath components. In the time domain, the channel impulse response (CIR), h_τ , is modelled as a filter with multiple taps, each tap represents a time-delay path of the transmitted signal as shown in Equation (5)

$$h_\tau = \sum_{p=1}^P a_p e^{-j\theta_p} \delta(\tau - \tau_p), \quad (5)$$

where a_p , θ_p , and τ_p are the complex amplitude, phase and delay of the p th path, respectively, P is the number of multipaths, and δ is the Dirac delta function.

In the frequency domain, the Fourier transform of the CIR yields the channel frequency response (CFR), H , which provides information about the amplitude and phase of the subcarriers, as shown in Equation (6)

$$H_i = |H_i| e^{j\phi_i}, \quad (6)$$

where H_i is the complex frequency response of the i th subcarrier, with $|H_i|$ and ϕ_i being the magnitude and phase of the i th subcarrier, respectively.

By applying signal processing techniques, the TOA and the AOA can be extracted from the CSI, which can be used to generate the range and angle information for the range-based localization methods. In addition, since the CSI provides a nearly complete radio profile for a specific location, the target's location can be found by performing a pattern matching of the CSI values.

In this work, the DNN models are trained using the RSS, AOA, and CSI measurements. The obtained root-mean-square error (RMSE) values are compared with the ones from the existing range-based hybrid RSS/AOA, where angle and RSS

measurements are used, and the range-free fingerprinting is based on RSS and CSI measurements. The detailed description of the existing hybrid RSS/AOA and fingerprinting used in the comparison can be found in Sections 3 and 4, respectively.

3 | EXISTING RANGE-BASED LOCALIZATION

In the range-based approach, signal parameters, such as the RSS and AOA, are first transformed into the range and angle information through (3) and (4), respectively. A set of LOP equations with respect to the obtained ranges and angles are created, and the target's location is found by solving these LOP equations using methods such as least square (LS) or subspace (SS). A review of related range-based methods is given below.

3.1 | RSS-least square

We first consider the case where only the RSS or range information is observed. For a general localization, we assume that there are a total of N anchors and M targets in the system. The anchors are placed at known locations $\mathbf{a}_n = [a_{nx}, a_{ny}]$ while the targets are at locations $\mathbf{b}_m = [b_{mx}, b_{my}]$, where $N = 1, \dots, N$ and $M = 1, \dots, M$, respectively. The range between any anchor and target pair can be found from Equation (3) as

$$r_{m,n} = \|\mathbf{b}_m - \mathbf{a}_n\| = r_0 10^{\frac{\text{RSS}_{a_n} - \text{RSS}_n + w_{a_n}}{10\gamma}}, \quad (7)$$

where $r_{m,n}$ is the range between the n th anchor and m th target, and RSS_n is the RSS at the n th anchor, and $\|\cdot\|$ is the Euclidean distance between \mathbf{a}_n and \mathbf{b}_m .

From the range information, a circular LOP with the radius of $r_{m,n}$ is formed around the n th anchor. In an ideal environment where there is no obstruction between the anchor and the target, the intersection of the two circular LOPs obtained from two anchors is enough to accurately determine the target's location. However, in a real-world environment, this situation rarely happens, thus at least three anchors are required.

Without loss of generality, the problem of locating a single m th target using N anchors is considered. Based on Equation (7), N circular LOP equations are constructed as

$$(b_{mx} - a_{nx})^2 + (b_{my} - a_{ny})^2 = r_{m,n}^2, \quad (8)$$

for $n = 1, \dots, N$. The solution to this problem can be found through various methods. Here, a simple LS method is presented.

The problem in Equation (8) is first linearized by some mathematical manipulations. As proposed in Ref. [25], the o th anchor located at $\mathbf{a}_o = [a_{ox}, a_{oy}]$ is initially selected as the reference anchor. Then, the remaining $N - 1$ LOP equations are subtracted from the LOP of the reference anchor. The results are as shown below

$$\begin{aligned} & (a_{nx} - a_{ox})b_{mx} + (a_{ny} - a_{oy})b_{my} \\ &= \frac{1}{2} \left[a_{nx}^2 + a_{ny}^2 - a_{ox}^2 - a_{oy}^2 - r_{m,n}^2 + r_{m,o}^2 \right], \end{aligned} \quad (9)$$

where $n = 1, \dots, N$ and $n \neq o$. In a matrix form, Equation (9) can be written as $\mathbf{A}_r \mathbf{b}_m^\top = \mathbf{C}_r$, where

$$\mathbf{A}_r = \begin{bmatrix} a_{1x} - a_{ox} & a_{1y} - a_{oy} \\ \vdots & \vdots \\ a_{Nx} - a_{ox} & a_{Ny} - a_{oy} \end{bmatrix}, \quad (10)$$

and

$$\mathbf{C}_r = \frac{1}{2} \begin{bmatrix} a_{1x}^2 + a_{1y}^2 - a_{ox}^2 - a_{oy}^2 - r_{m,1}^2 + r_{m,o}^2 \\ \vdots \\ a_{Nx}^2 + a_{Ny}^2 - a_{ox}^2 - a_{oy}^2 - r_{m,N}^2 + r_{m,o}^2 \end{bmatrix}. \quad (11)$$

By using the LS method, the estimated location of the m th target, $\tilde{\mathbf{b}}_m$, can be found from Equation (12)

$$\tilde{\mathbf{b}}_m^\top = (\mathbf{A}_r^\top \mathbf{A}_r)^{-1} \mathbf{A}_r^\top \mathbf{C}_r. \quad (12)$$

Although the problem is much simplified by linearization, the accuracy obtained from the linear LS method is generally lower than the one obtained from its non-linear counterpart.

3.2 | RSS-subspace

In the subspace method, a certain amount of noise is removed from the observed RSS through a subspace projection. Therefore, the subspace offers higher localization accuracy compared with the RSS-LS method. However, it also requires higher computational effort. Details of the signal projection are described below.

As proposed in Ref. [44], a symmetric matrix \mathbf{D} is first defined as

$$\mathbf{D} = \mathbf{X} \mathbf{X}^\top, \quad (13)$$

where

$$\mathbf{X} = \begin{bmatrix} a_{1x} - b_{mx} & a_{1y} - b_{my} \\ \vdots & \vdots \\ a_{Nx} - b_{mx} & a_{Ny} - b_{my} \end{bmatrix}. \quad (14)$$

Clearly, \mathbf{D} and \mathbf{X} have the rank of 2. The elements D_{ij} can be found as

$$\begin{aligned} D_{ij} &= (a_{ix} - b_{mx})(a_{jx} - b_{mx}) + (a_{iy} - b_{my})(a_{jy} - b_{my}) \\ &= \frac{1}{2} [d_i^2 + d_j^2 - d_{ij}^2], \end{aligned} \quad (15)$$

where d_i and d_j are the real distances from the i th and the j th anchors to the m th target respectively, and d_{ij} is the real distance between the i th and the j th anchors. By using eigenvalue decomposition, \mathbf{D} can be written as

$$\mathbf{D} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \quad (16)$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues $\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N]$ is a matrix whose columns are the orthonormal eigenvectors corresponding to the eigenvalues. Since \mathbf{D} is a rank-2 matrix, we have $\lambda_3 = \lambda_4 = \dots = \lambda_N = 0$. By defining the subspace matrices $\mathbf{\Lambda}_s = \text{diag}(\lambda_1, \lambda_2)$ and $\mathbf{U}_s = [\mathbf{u}_1, \mathbf{u}_2]$, we have

$$\begin{aligned} \mathbf{D} &= \mathbf{U}_s \mathbf{\Lambda}_s \mathbf{U}_s^T \\ &= \mathbf{U}_s \mathbf{\Lambda}_s^{1/2} (\mathbf{U}_s \mathbf{\Lambda}_s^{1/2})^T \\ &= \mathbf{U}_s \mathbf{\Lambda}_s^{1/2} \mathbf{\Omega} (\mathbf{U}_s \mathbf{\Lambda}_s^{1/2} \mathbf{\Omega})^T, \end{aligned} \quad (17)$$

where $\mathbf{\Omega}$ is a rotational matrix satisfying $\mathbf{\Omega}\mathbf{\Omega}^T = \mathbf{I}_N$, that is, identity matrix of order N . From Equations (13) and (17), we have

$$\mathbf{X} = \mathbf{U}_s \mathbf{\Lambda}_s^{1/2} \mathbf{\Omega}, \quad (18)$$

and an estimate of $\mathbf{\Omega}$ obtained using the LS method is

$$\mathbf{\Omega} = \left[(\mathbf{U}_s \mathbf{\Lambda}_s^{1/2})^T (\mathbf{U}_s \mathbf{\Lambda}_s^{1/2}) \right]^{-1} (\mathbf{U}_s \mathbf{\Lambda}_s^{1/2})^T \mathbf{X}. \quad (19)$$

Then, by substituting Equation (19) back into Equation (18), we obtain

$$\mathbf{X} = \mathbf{U}_s \mathbf{U}_s^T \mathbf{X}. \quad (20)$$

Here, the subspace matrix \mathbf{U}_s is obtained from the matrix \mathbf{D} whose elements are generated from the real distances. However, in practice, \mathbf{D} cannot be obtained directly since we have no knowledge about the distances d_i and d_j . Therefore, an estimate of \mathbf{D} , denoted as $\tilde{\mathbf{D}}$, is used instead, and its elements can be found as

$$\tilde{D}_{i,j} = \frac{1}{2} [r_i^2 + r_j^2 - d_{ij}^2], \quad (21)$$

where r_i and r_j are the approximate distances from the i th and the j th anchors, respectively. Similarly, if we define $\tilde{\mathbf{U}}_s$ as a subspace obtained from $\tilde{\mathbf{D}}$, we have an approximation

$$\mathbf{X} \approx \tilde{\mathbf{U}}_s \tilde{\mathbf{U}}_s^T \mathbf{X}. \quad (22)$$

To find the location of the target, \mathbf{b}_m , we first define matrix \mathbf{A} as

$$\mathbf{A} = \begin{bmatrix} a_{1x} & a_{1y} \\ \vdots & \vdots \\ a_{Nx} & a_{Ny} \end{bmatrix}, \quad (23)$$

then from Equation (14), \mathbf{X} can be written as

$$\mathbf{X} = \mathbf{A} - \mathbf{1}_N \mathbf{b}_m, \quad (24)$$

where $\mathbf{1}_N$ is an $N \times 1$ column vector with all elements equal to 1. By using the LS method, an estimate of \mathbf{b}_m is

$$\hat{\mathbf{b}}_m^T = \frac{\mathbf{1}_N^T (\mathbf{I}_N - \tilde{\mathbf{U}}_s \tilde{\mathbf{U}}_s^T) \mathbf{A}}{\mathbf{1}_N^T (\mathbf{I}_N - \tilde{\mathbf{U}}_s \tilde{\mathbf{U}}_s^T) \mathbf{1}_N}. \quad (25)$$

3.3 | Hybrid RSS/AOA least square

Next, we consider the case where the AOA or the angle information is observed in addition to the RSS. In Figure 3, the shaded area represents the possible locations of the target based on the RSS measurements. After the AOA measurements are added to the calculation, the shaded area becomes smaller, as shown in the diagonally-shaded area.

We assume that N RSS and N AOA measurements are obtained from N anchors. From this information, N linear equations for the target's x - and y -axis coordinates can be constructed as

$$\begin{aligned} \tilde{b}_{mx} &= a_{nx} + r_{m,n} \cos \alpha_n \\ \tilde{b}_{my} &= a_{ny} + r_{m,n} \sin \alpha_n, \end{aligned} \quad (26)$$

for $n = 1, \dots, N$. In a matrix form, Equation (26) can be written as $\mathbf{E} \tilde{\mathbf{b}}_m = \mathbf{C}_a$ where

$$\mathbf{E} = \begin{bmatrix} \mathbf{1}_N & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{1}_N \end{bmatrix}, \quad (27)$$

and

$$\mathbf{C}_a = \begin{bmatrix} a_{1x} + r_{m,1} \cos \alpha_1 \\ \vdots \\ a_{Nx} + r_{m,N} \cos \alpha_N \\ a_{1y} + r_{m,1} \sin \alpha_1 \\ \vdots \\ a_{Ny} + r_{m,N} \sin \alpha_N \end{bmatrix}. \quad (28)$$

The estimate location of the target obtained from the LS method is then

$$\hat{\mathbf{b}}_m^T = (\mathbf{E}^T \mathbf{E})^{-1} \mathbf{E}^T \mathbf{C}_a. \quad (29)$$

3.4 | Hybrid RSS/AOA weighted least squares

Since the anchors closer to the target are likely to provide a smaller estimation error, Ref. [27] proposed a weighted LS

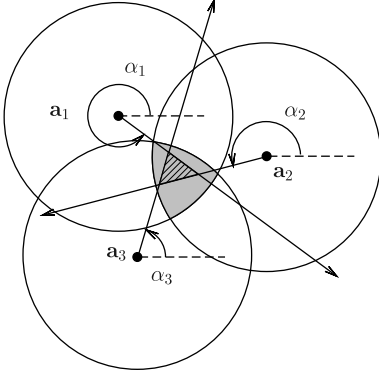


FIGURE 3 Hybrid received signal strength/angle-of-arrival localization system

method, where the ranges obtained from the anchors closer to the target are weighted higher, that is,

$$\tilde{\mathbf{b}}_m^\top = (\mathbf{E}^\top \mathbf{W}^\top \mathbf{E})^{-1} \mathbf{E}^\top \mathbf{W}^\top \mathbf{C}_a, \quad (30)$$

where \mathbf{E} and \mathbf{C}_a are the matrices described in Equations (27) and (28). The weight matrix, \mathbf{W} , is defined as

$$\mathbf{W} = \mathbf{I}_2 \otimes \text{diag}\{\mathbf{w}\}, \quad (31)$$

where \mathbf{I}_2 is an identity matrix of order 2, \otimes is the Kronecker product, $\mathbf{w} = [\sqrt{w_1}, \sqrt{w_2}, \dots, \sqrt{w_N}]$, and $w_n = 1 - (r_{m,n} / \sum_{n=1}^N r_{m,n})$. Results show that the weighted least squares (WLS) method is able to achieve higher localization accuracy, compared with the standard LS method.

3.5 | Unbalanced hybrid RSS/AOA

In previously mentioned methods, N RSS and N AOA values are required in the calculations. This means that every anchor must be equipped with an antenna array for AOA measurement. For a system with limited resources, such as wireless sensor networks, installing an antenna array on every sensor node would add a large amount of complexity to the system. Therefore, Ref. [30] proposed an unbalanced hybrid RSS/AOA localization, where only the master anchor is responsible for the AOA measurement. As a result, the information obtained from all N anchors consists of N RSS and 1 AOA values.

To combine 1 AOA measurement with N RSS measurements, the authors used a simple range projection method, as shown in Figure 4. At the master anchor \mathbf{a}_a , range r_a and angle α_a are used to create two virtual ranges r_{v1} and r_{v2} corresponding to the two virtual anchors at \mathbf{a}_{v1} and \mathbf{a}_{v2} by

$$\begin{aligned} \mathbf{a}_{v1} &= \mathbf{a}_a + [r_a \cos \alpha_a, 0], \\ \mathbf{a}_{v2} &= \mathbf{a}_a + [0, r_a \sin \alpha_a], \\ r_{v1} &= r_a \sin \alpha_a, \\ r_{v2} &= r_a \cos \alpha_a. \end{aligned} \quad (32)$$

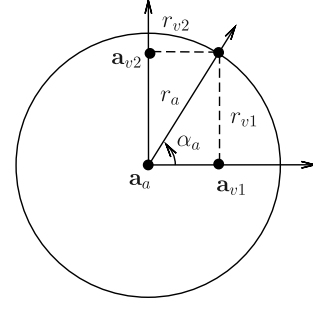


FIGURE 4 Virtual anchors obtained from one received signal strength and one angle-of-arrival measurement

Thus, we have a total of $N + 1$ anchors and their $N + 1$ range measurements. The location of the target can then be determined using the LS method or the SS method as previously described. In addition, to further improve the localization accuracy, Ref. [30] proposed that the anchor with the smallest distance to the target should be selected as the master anchor.

In theory, the Cramer–Rao lower bound (CRLB) indicates the performance limit of an unbiased estimator. The CRLB of the hybrid RSS/AOA method is given in Ref. [30]. In addition to the RSS, the CSI has been used for the range-based localization as well. By applying signal processing techniques such as the multiple signal classification (MUSIC) algorithm on the obtained CSI, the TOA and AOA can be extracted without the need for a dedicated equipment. The CRLB for the range-based TOA/AOA CSI localization is presented in the Appendix. However, an extended information and experimentation on the hybrid TOA/AOA localization based on the CSI measurement are out of scope of this paper.

Since the range-based methods usually locate the target in the geometric domain, its accuracy depends highly on the accuracy of the range and angle measurements, which in turn depends on the existence of LOS paths.

4 | EXISTING RANGE-FREE LOCALIZATION

Another approach to wireless localization is fingerprinting. The method is based on fingerprint matching. Unlike the range-based approach, the LOS path is not a requirement. In this section, we cover some of the well-known fingerprinting methods, that is, RADAR, Horus, and Fine-grained Indoor Fingerprint System (FIFS).

4.1 | RADAR

RADAR [31] is one of the earliest fingerprinting methods for indoor localization. The RADAR system uses three anchors to collect the RSS information to create the radio map. Let \mathcal{X} be the set of reference locations, three average RSS values obtained from the three anchors are stored as the fingerprint for each $x \in \mathcal{X}$.

To determine the location of the target, the distance between the real-time fingerprint and the ones in the radio map are calculated. We define d_x as the distance between the real-time fingerprint and the reference fingerprint at the location x , s_i as the real-time RSS value obtained from the i th anchor, and $f_{i,x}$ as the reference RSS value obtained from the i th anchor at the location x . The Euclidean distance between the real-time and the reference fingerprints, d_x , is defined as

$$d_x = \sqrt{(s_1 - f_{1,x})^2 + (s_2 - f_{2,x})^2 + (s_3 - f_{3,x})^2}, \quad (33)$$

for all $x \in \mathcal{X}$. The estimated target's location is the location x , which minimizes d_x .

4.2 | Horus

Unlike the deterministic approach used in RADAR, a probabilistic approach is used in Horus [32]. We assume that there are N anchors covering a set of reference locations \mathcal{X} , the RSS distributions $P\{s_i|x\}$ for $i = 1, \dots, N$ are stored as the radio map for every location $x \in \mathcal{X}$. Given the real-time RSS readings from N anchors, $\mathbf{s} = \{s_1, \dots, s_N\}$, the estimated target's location \tilde{x} is the location that maximises the probability $P\{x|\mathbf{s}\}$, or mathematically

$$\tilde{x} = \arg \max_x P\{x|\mathbf{s}\} = \arg \max_x P\{\mathbf{s}|x\}, \quad (34)$$

where the second equality comes from an assumption that the target is equally likely to be in any location. The conditional probability $P\{\mathbf{s}|x\}$ can be computed using information from the radio map as

$$P\{\mathbf{s}|x\} = \prod_{i=1}^N P\{s_i|x\}. \quad (35)$$

Based on Equation (34) and (35), \tilde{x} is approximately one of the reference locations. To improve an accuracy, weighted average of the location can be used.

4.3 | Fine-grained Indoor Fingerprint System

FIFS [34] uses a probabilistic approach similar to Horus; however, the CSI is used instead of the RSS. A radio map is created using the distributions of the effective CSI from N anchors, which is defined as

$$H_{e,n} = \sum_{i=1}^{N_{sc}} |H_{i,n}|^2, \quad (36)$$

where $H_{e,n}$ is the effective CSI of the n th anchor, $H_{i,n}$ is the CSI for the i th subcarrier of the n th anchor, and N_{sc} is the total number of the subcarriers.

During the online phase, given a vector of the real-time effective CSI, $\mathbf{H}_e = \{H_{e,1}, \dots, H_{e,N}\}$, the estimated target's location \tilde{x} is the location $x \in \mathcal{X}$ that maximises the posterior probability $P\{x|\mathbf{H}_e\}$, which is defined as

$$P\{x|\mathbf{H}_e\} = \frac{P\{x\}P\{\mathbf{H}_e|x\}}{\sum_{x \in \mathcal{X}} P\{x\}P\{\mathbf{H}_e|x\}} = \frac{P\{x\}P\{\mathbf{H}_e|x\}}{P\{\mathbf{H}_e\}}. \quad (37)$$

Similar to Horus, using the posterior probability to perform a weighted average of the locations can be used to improve the accuracy.

5 | PROPOSED DNN-BASED LOCALIZATION

As the performance limits of the existing range-based and range-free localization methods come from the imprecision of the hand-designed geometric models and algorithms, in this work, the DNN approach is used to automatically extract the relationship between the observed signals and the target's location. We investigate the performance of several types of DNN models when they are trained on both the range-based and range-free datasets. The range-based data sample is formatted as shown below.

$$(\text{input signal parameters, outputs: } b_{mx}, b_{my}), \quad (38)$$

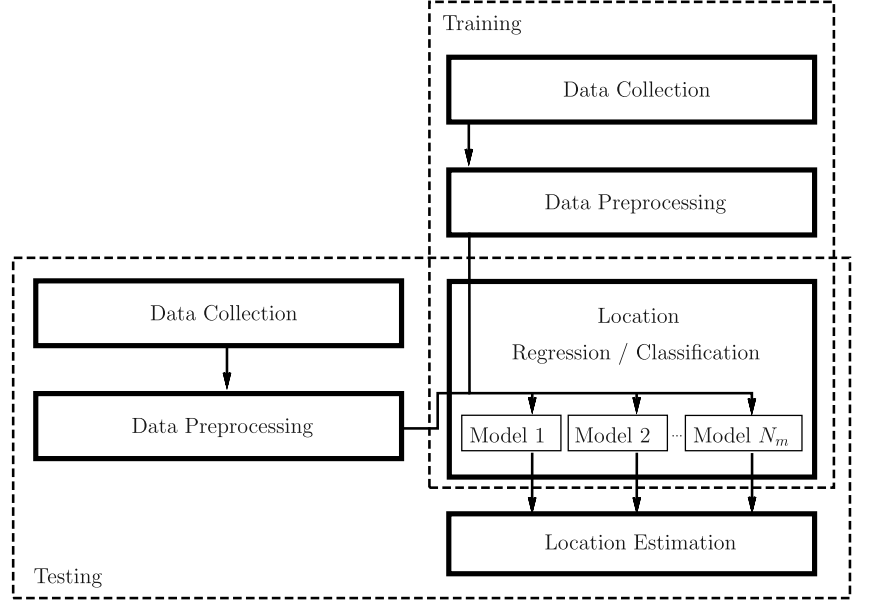
where the input parameters consist of a number of the RSS and AOA measurements from the anchors, b_{mx} and b_{my} are the true target's location in terms of x and y -axis coordinates. The range-free data sample is formatted as

$$(\text{input: signal parameters, output: reference location ID}), \quad (39)$$

where the input parameters are either the RSS or the CSI measurements, and the output location ID is a unique identification number for each reference location. For the range-based dataset, the problem is modelled as a regression task, whereas for the range-free dataset, it is modelled as a classification task. The structure of the proposed DNN-based localization for the two tasks is shown in Figure 5.

There are four main units in the proposed system: data collection, data pre-processing, location regression/classification, and location estimation. In the training phase, the RSS, AOA and CSI are first collected by the data collection unit. Then, the data are filtered and sanitized by the data pre-processing unit. Next, N_m base models are trained to produce a set of N_m outputs, where the outputs are the target's coordinates for the location regression unit, and the probability of the target being in each of the reference locations for the location classification unit, respectively. In the testing phase, the real-time RSS, AOA, and CSI measurements are collected, calibrated, and then passed onto the trained models where the outputs obtained from the N_m models are combined using the

FIGURE 5 Structure of the proposed deep-neural-network-based localization system



ensemble method to produce the final target's location in the location estimation unit. The detailed description of the units is presented below.

5.1 | Data collection unit

The data collection unit is responsible for collecting the RSS, AOA, and CSI measurements. To generate the range-based dataset, a system of N ($N \geq 3$) anchors is considered, and the RSS and AOA obtained from the N anchors are generated according to Equations (3) and (4). To generate the range-free dataset, a system with a single anchor is considered, and the RSS and CSI are collected from a real environment in our Telecommunications building.

To measure the RSS and CSI, a commodity 2.4 GHz Wi-Fi access point (AP) is used as the anchor while the target is a laptop equipped with the Intel Wi-Fi Link 5300 network interface card (IWL5300-NIC). The IWL5300-NIC is running on a modified driver developed by the authors in Ref. [45], which allows the laptop to extract the RSS and CSI from the packets transmitted by the anchor. According to the IEEE 802.11n-2009 standard [46], 56 subcarriers are used in the 20-MHz band. However, the IWL5300-NIC only supports the measurement of 30 subcarrier groups, which corresponds to a grouping number of 2 as defined by the standard. The CSI obtained from the IWL5300-NIC is normalized such that there is unit noise presented at the receiver. Generally, for an anchor with N_t transmit antennas and a target with N_r receive antennas, $N_r \times N_t \times 30$ complex CSI values are obtained from each packet.

5.2 | Data pre-processing unit

Data pre-processing is a necessary step prior to model training and testing as it helps to reduce the learning time and

complexity of the model. A moving filter defined in Equation (40) is used for the RSS and AOA while a three-step filtering method is proposed for the CSI.

$$y_i = \frac{1}{W} \sum_{j=0}^{W-1} x_{i+j}, \quad (40)$$

where y is the filtered output, x is the input, and W is the number of points used in the calculation.

The CSI obtained from the 2.4-GHz anchor may contain some inconsistent readings due to the interference from other 2.4-GHz Wi-Fi APs, which can be seen as large drops in the magnitude response of the CSI in some packets. Figure 6 shows a surface plot of normalized CSI magnitudes from 200 consecutive packets when the receiver is placed at a fixed location. To sanitize the CSI, three steps consisting of multipath filtering, outlier filtering, and phase calibration are used.

5.2.1 | Multipath filtering

In multipath filtering, the effect of insignificant paths presented in the CIR is removed. To achieve this, the CSI is first converted into a time-domain CIR using the inverse Fourier transform, as shown in Figure 7. From the CIR, only the first l channel taps are considered, and the rest of the channel taps are filtered out. The filtered CIR is converted back to the frequency domain using the Fourier transform to obtain the filtered CSI.

5.2.2 | Outlier filtering

Next, the outliers in the filtered CSI are removed. For each packet, an average CSI magnitude is computed over the 30

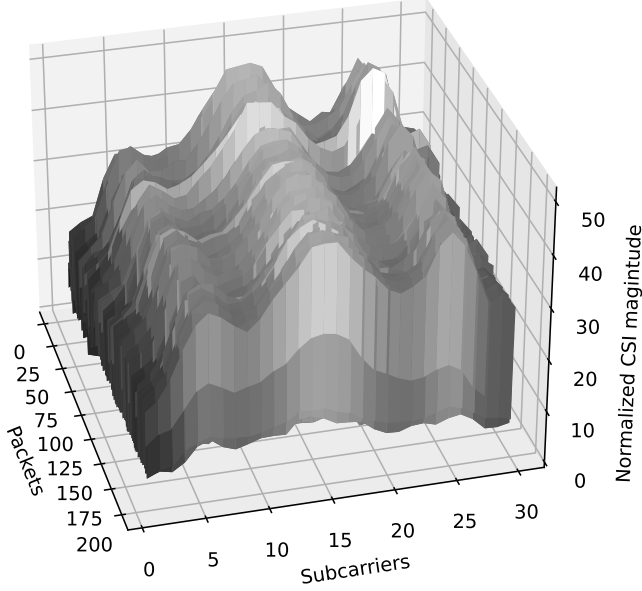


FIGURE 6 A surface plot of normalized channel state information magnitudes of 200 consecutive packets for a single location

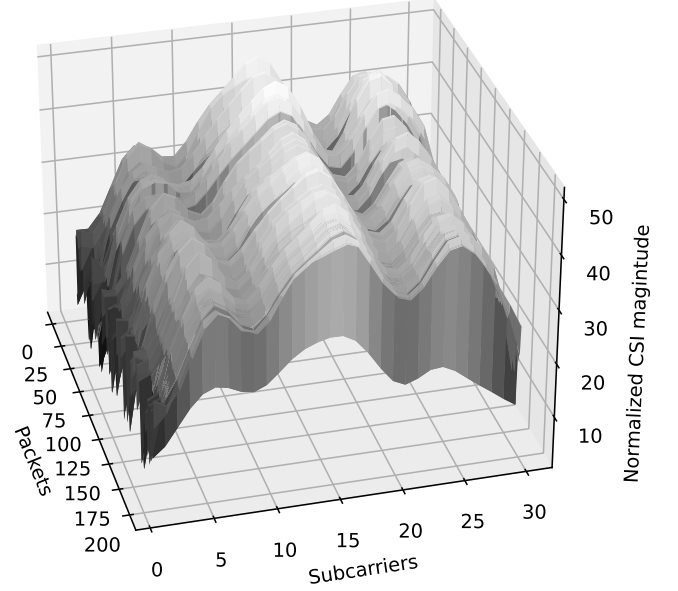


FIGURE 8 A surface plot of filtered channel state information (CSI) readings from 200 packets when the receiver is placed at a fixed location

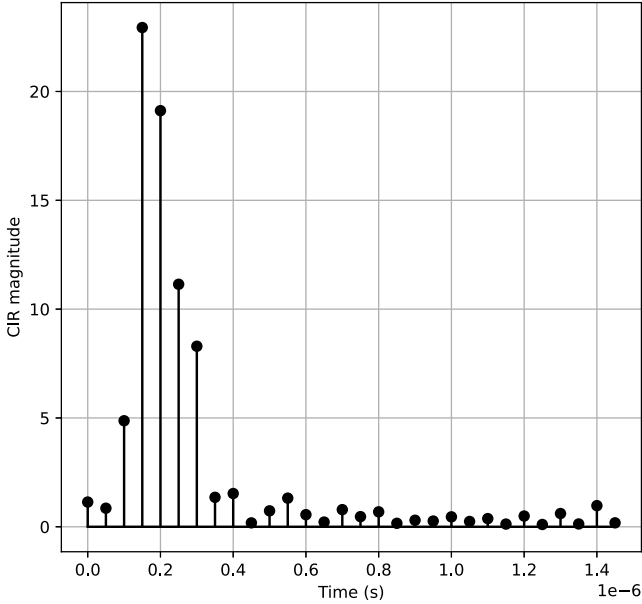


FIGURE 7 Time-domain channel impulse response (CIR)

subcarrier groups. Then, using a sliding of size w packets, only the packet with the highest average magnitude within the window is selected. This is based on the idea that a higher CSI magnitude indicates a more reliable packet reading.

Let \mathcal{N}_p be the set of valid packet indices, w be the size of the sliding window (in packets), and $|H|_{n,i}$ be the magnitude of the complex CSI $H_{n,i}$ of the i th subcarrier from the n th packet. The algorithm for outlier filtering is summarised by a pseudocode in Algorithm 1. The outlier filtering result in a more consistent CSI is shown in Figure 8.

Algorithm 1 Outlier filtering

```

1: set  $\mathcal{N}_p = \emptyset$ 
2: set  $w$ 
3: set  $c = 0$ 
4: while  $c + w \leq \text{number of packets available}$ 
do
5:   for  $n = 1$  to  $w$  do
6:     collect  $H_{c+n,i} \forall i \in \{1, \dots, 30\}$ 
7:     calculate  $|H|_{c+n} = \frac{1}{30} \sum_{i=1}^{30} |H|_{c+n,i}$ 
8:   end for
9:   find  $n^* = \arg \max_n |H|_{c+n}$ 
10:  update  $\mathcal{N}_p \leftarrow \mathcal{N}_p \cup n^*$ 
11:   $c = c + 1$ 
12: end while

```

5.2.3 | Phase calibration

Although the filtered CSI magnitudes are more consistent, their phases are still random due to the fact that the AP and the receiver are not synchronized. The dash lines in Figure 9 show the raw CSI phases (after unwrapping) of a subcarrier over 200 packets.

To calibrate the phase, a linear transformation as proposed in Ref. [47, 48] is used. Let $\hat{\phi}_i$ be the measured (raw) phase of the i th subcarrier, which is defined as

$$\hat{\phi}_i = \phi_i - 2\pi \frac{k_i}{N_{\text{FFT}}} \delta + \beta + Z, \quad (41)$$

where ϕ_i is the true phase of the i th subcarrier, k_i is the subcarrier index ($-28, \dots, 28$ in the 20-MHz operating mode),

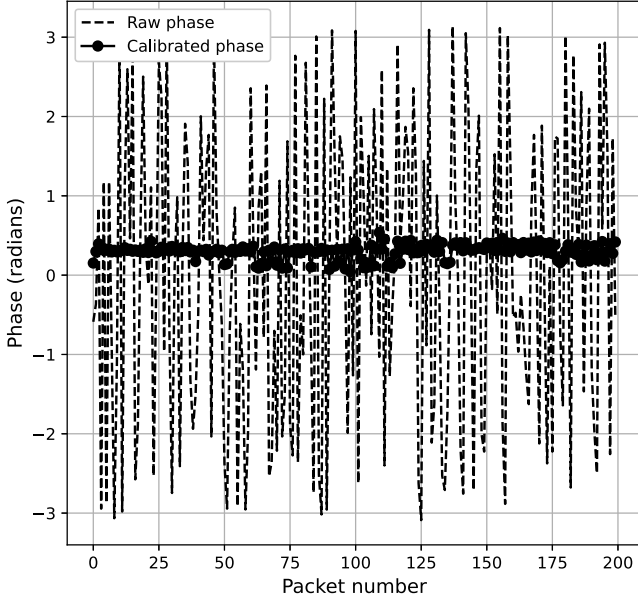


FIGURE 9 Channel state information phase of a subcarrier over 200 packets

N_{sc} is the total number of subcarriers, $N_{FFT} = 64$ is the FFT size as defined in the IEEE 802.11 a/g/n, δ is the timing offset at the receiver, β is the unknown phase offset, and Z is the measurement noise. Hence, removing the unknown δ and β terms would stabilise the phase. First, we define the parameters a and b as

$$a = \frac{\hat{\phi}_{N_{sc}} - \hat{\phi}_1}{k_{N_{sc}} - k_1}, \quad (42)$$

$$b = \frac{1}{N_{sc}} \sum_{i=1}^{N_{sc}} \hat{\phi}_i, \quad (43)$$

where the calibrated phase of the i th subcarrier, $\tilde{\phi}_i$, is obtained by performing a linear combination of the true phases as shown below

$$\begin{aligned} \tilde{\phi}_i &= \hat{\phi}_i - (ak_i + b) \\ &= \phi_i - \frac{\phi_{N_{sc}} - \phi_1}{k_{N_{sc}} - k_1} k_i - \frac{1}{N_{sc}} \sum_{i=1}^{N_{sc}} \phi_i + \frac{1}{N_{sc}} \frac{2\pi\delta}{N_{FFT}} \sum_{i=1}^{N_{sc}} k_i \\ &\approx \phi_i - \frac{\phi_{N_{sc}} - \phi_1}{k_{N_{sc}} - k_1} k_i - \frac{1}{N_{sc}} \sum_{i=1}^{N_{sc}} \phi_i, \end{aligned} \quad (44)$$

where we assume that measurement noise is negligible, and $\sum_{i=1}^{N_{sc}} k_i = 0$. The calibrated phase is shown in Figure 9.

After the magnitude and phase of the RSS and CSI measurement are processed, the data are passed onto the location regression/classification unit.

5.3 | Location regression/classification unit

The location regression/classification unit roughly determines the location of the target. For the range-based dataset, a regression model is used, whereas, for the range-free dataset, a classification model is used. Within this unit, a base DNN model is trained N_m times using the same set of training data, but with different initial conditions, thus generating N_m models that have the same architecture with a different set of weights. Several types of DNN models are considered for the base model, including the FNN, 1D-CNN, 2D-CNN and LSTM. An overview of the DNN models is provided below.

5.3.1 | Feed-forward neural network

FNN is a multi-layer network of neurons, where information are passed only in the forward direction. The mechanism of a neuron is shown in Figure 10. Here g_1, g_2, \dots, g_N are the weights corresponding to the inputs x_1, x_2, \dots, x_N , respectively, b is the bias term, and $\varphi(\cdot)$ is the activation function. To compute the output y , the inputs are multiplied by their weights, then summed with the bias, and fed into the activation function, as described below

$$y = \varphi(g_1x_1 + g_2x_2 + \dots + g_Nx_N + b). \quad (45)$$

For the FNN to be able to estimate non-linear mapping between the inputs and outputs, non-linear activation functions, such as sigmoid, hyperbolic tangent, and rectified linear unit (ReLU), are used. For more complicated models, the neurons are connected into layers consisting of an input layer, several hidden layers, and an output layer, as shown in Figure 11. Based on this structure, the outputs of the prior layer are the inputs to the subsequent layers and so on, until the final output layer is reached. In addition, the order of the data is not considered in this type of model.

5.3.2 | 2D-convolutional neural network

2D-CNN is a neural network designed to work with 2D inputs such as images. The architecture of the 2D-CNN consists of an automatic feature extraction unit stacking on top of the FNN, as shown in Figure 12. The feature extraction unit contains two types of layers called a convolutional layer and a pooling layer. The convolutional layer uses a number of filters to perform convolutional filtering on the input data and creating the outputs called feature maps. By stacking multiple convolutional layers on top of each other, higher-order features, such as lines and shapes, can be extracted. The pooling layer behaves as a down-sampling layer. It compresses an input feature map by taking the average or the maximum of the inputs. The use of convolutional layer, pooling layer, and FNN allows this type of neural network to be resilient to input transformations, such as translation, rotation, and scaling.

5.3.3 | 1D-convolutional neural network

1D-CNN is a variant of the 2D-CNN designed to work with 1D inputs, such as time series. The architecture of the 1D-CNN is the same as the one presented in the 2D-CNN, with an automatic feature extraction stacking on top of the FNN. The 1D-CNN also share similar operations as the 2D-CNN; however, the convolutional filters now cover the entire length of the features and move in only one direction, as shown in Figure 13. Much like the 2D-CNN, the 1D-CNN is also resilient to the 1D input transformations.

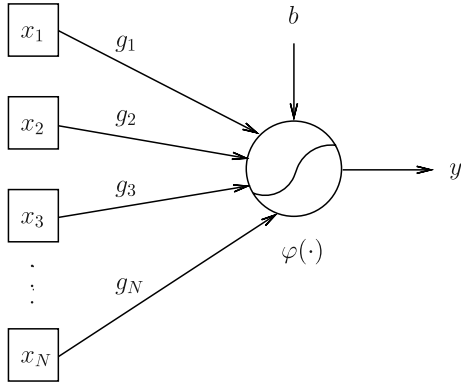


FIGURE 10 A single neuron

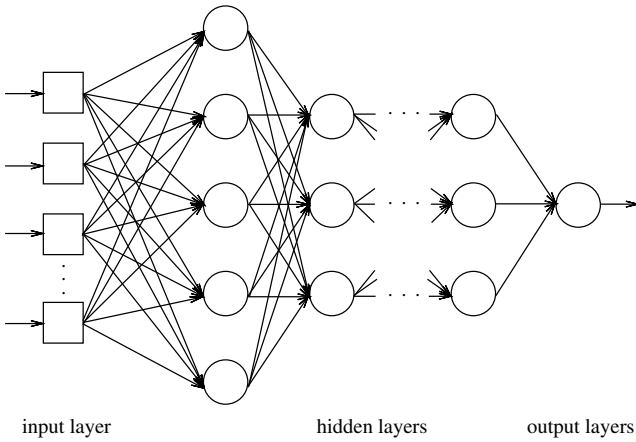


FIGURE 11 Feed-forward neural network

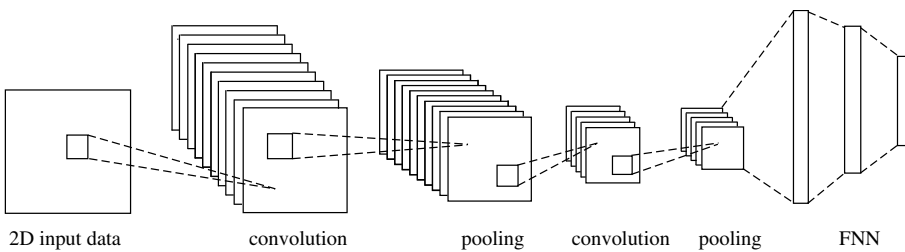


FIGURE 12 2D-convolutional neural network

5.3.4 | Long short-term memory network

LSTM is a type of recurrent neural networks, which contains loops in them, as shown in Figure 14. These loops allow the network to memorize prior input sequences, making them suitable for tasks with sequential inputs, such as time-series prediction and speech recognition. Unrolling the loop, an LSTM is a chain of multiple LSTM cells, where each cell contains a cell state \mathbf{Q} , which carries information through an entire chain and a mechanism to decide which piece of information should be passed along or dropped from the chain. This mechanism is controlled by 3 types of gates, that is, forget gate, input gate, and output gate, as shown in Figure 15.

The forget gate decides which piece of information is to be kept or removed from the cell state \mathbf{Q} . Given the current input sequence \mathbf{x}_t and the output of the previous hidden state \mathbf{h}_{t-1} , the output \mathbf{f}_t is expressed as

$$\mathbf{f}_t = \sigma(\mathbf{w}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f), \quad (46)$$

where \mathbf{w}_f and \mathbf{b}_f are the weights and the biases related to the forget gate, respectively. A zero in \mathbf{f}_t indicates the part of the previous cell state \mathbf{Q}_{t-1} that should be dropped, while a one indicates the part that should be kept.

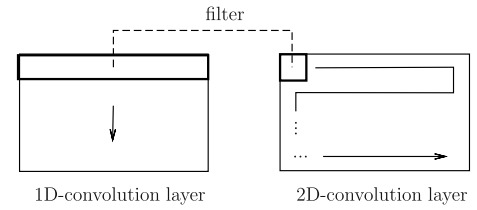


FIGURE 13 Comparison between 1D-convolutional neural network and 2D-convolutional neural network

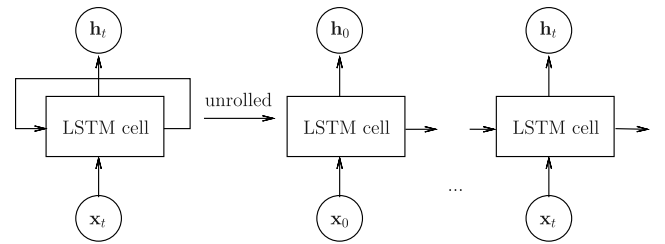


FIGURE 14 A long short-term memory network (LSTM) cell with an input \mathbf{x}_t and an output \mathbf{h}_t

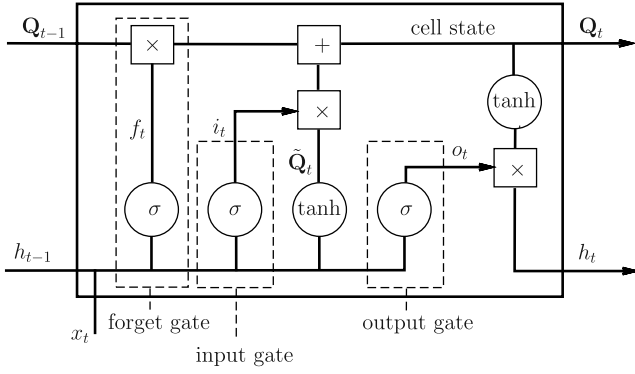


FIGURE 15 A long short-term memory network cell

The input gate decides which parts of the input should be stored in the cell state. From the diagram, we have

$$\mathbf{i}_t = \sigma(\mathbf{w}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (47)$$

$$\tilde{\mathbf{Q}}_t = \tanh(\mathbf{w}_q[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_q),$$

where \mathbf{w}_i , \mathbf{b}_i , \mathbf{w}_q and \mathbf{b}_q are the weights and biases related to the sigmoid function, and the weights and biases related to the hyperbolic tangent function, respectively. Then, the operation to update \mathbf{Q}_{t-1} to \mathbf{Q}_t is performed as shown below

$$\mathbf{Q}_t = \mathbf{f}_t \mathbf{Q}_{t-1} + \mathbf{i}_t \tilde{\mathbf{Q}}_t. \quad (48)$$

The output gate decide which part of the information should be outputted as \mathbf{h}_t . The process is described by

$$\begin{aligned} \mathbf{o}_t &= \sigma(\mathbf{w}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o), \\ \mathbf{h}_t &= \mathbf{o}_t \tanh(\mathbf{Q}_t), \end{aligned} \quad (49)$$

where \mathbf{h}_t is the output of the current hidden unit, \mathbf{Q}_t is the current cell state, and \mathbf{w}_o and \mathbf{b}_o are the weights and the biases related to the output gate.

We have seen that different types of DNN models offer different benefits, for example, the FNN can be applied to any type of data, the 2D-CNN and 1D-CNN offer automatic feature extraction, and LSTM can process sequential inputs. Therefore, in wireless localization, these models can be used to process raw input data from different aspects.

5.4 | Location estimation unit

Since the training of the DNN models is stochastic in nature, we assume that the N_m models perform poorly in different regions of feature space, and by combining the outputs of the models, the accuracy should be improved as a result of model diversity. The location estimation unit combines the N_m outputs obtained from the location regression/classification unit into the target's final location. For the range-based dataset, the outputs of the N_m models are stacked to create a $N_m \times 2$ matrix \mathbf{B} .

$$\mathbf{B} = \begin{bmatrix} b_{1,x} & b_{1,y} \\ b_{2,x} & b_{2,y} \\ \vdots & \vdots \\ b_{N_m,x} & b_{N_m,y} \end{bmatrix}, \quad (50)$$

where $b_{i,x}$ and $b_{i,y}$ are the target's x - and y -axis coordinates estimated by the i th model. The target's location is simply an average of the outputs in matrix \mathbf{B} , that is,

$$\tilde{b}_{mx} = \frac{1}{N_m} \sum_{n=1}^{N_m} b_{n,x} \quad (51)$$

$$\tilde{b}_{my} = \frac{1}{N_m} \sum_{n=1}^{N_m} b_{n,y},$$

where \tilde{b}_{mx} and \tilde{b}_{my} are the estimated x - and y -axis coordinates of the target.

Similarly, for the range-free dataset, the outputs of the N_m models are stacked to create a $N_m \times L$ matrix \mathbf{V} .

$$\mathbf{V} = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,L} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N_m,1} & v_{N_m,2} & \cdots & v_{N_m,L} \end{bmatrix}, \quad (52)$$

where L is the total number of the reference locations, and the element $v_{i,j}$ is the posterior probability obtained from the i th model for the j th reference location. The target's location is the weighted average of the reference locations, where the weights are found by summing along the columns of the matrix \mathbf{V} . Let \mathbf{g} be the matrix of the weights, we have

$$\mathbf{g} = [g_1 \quad g_2 \quad \cdots \quad g_L], \quad (53)$$

where g_l is the weight for the l th reference location, which can be calculated from

$$g_l = \frac{\sum_{n=1}^{N_m} v_{n,l}}{\sum_{l=1}^L \sum_{n=1}^{N_m} v_{n,l}}. \quad (54)$$

Finally, the location of the m th target is the average of the reference location using the weights in \mathbf{g} as

$$\tilde{\mathbf{b}}_m = [\tilde{b}_{mx}, \tilde{b}_{my}] = \sum_{l=1}^L g_l \mathbf{x}_l, \quad (55)$$

where \mathbf{x}_l is the coordinate of the l th reference location.

6 | EXPERIMENTATION AND RESULTS

The experiment is separated into two parts based on the range-based and range-free datasets. The performance of the proposed and the existing methods in both experiments are evaluated in terms of the RMSE, as defined by

$$\text{RMSE} = \sqrt{\frac{1}{K} \sum_{k=1}^K d_{k,e}^2}, \quad (56)$$

where K is the total number of samples, and $d_{k,e}$ is the distance error for the k th sample, which is given by

$$d_{k,e} = \sqrt{(b_{k,mx} - \tilde{b}_{k,mx})^2 + (b_{k,my} - \tilde{b}_{k,my})^2}, \quad (57)$$

where $b_{k,mx}$ and $b_{k,my}$ are true coordinates of the m th target for the k th sample, $\tilde{b}_{k,mx}$ and $\tilde{b}_{k,my}$ are the estimated coordinates of the m th target for the k th sample.

6.1 | Experimentation on the range-based dataset

In our first experiment, the range-based dataset is considered. The training and testing samples are formatted as shown in Equation (38), where the input parameters consist of anchors' locations, observed ranges, and observed angles while the outputs are the target's coordinates.

6.1.1 | Simulation setting

We consider the case where the number of anchors $N \geq 3$ are used to locate a single target. The location of the anchors and the target are generated randomly according to a uniform distribution within an area of $50 \times 50 \text{ m}^2$. The observed ranges and angles follow the relationship given in Equations (4) and (7), respectively.

Since the samples are generated independently from sample to sample, a FNN model is used as the base model in the location regression unit. The structure of the model is determined through a series of experiments. The result is shown in Table 2. The model consists of an input layer, five hidden layers, and an output layer. The information in the bracket in each layer are the number of neurons and the activation function, respectively. Here, a dropout layer is added as a

TABLE 2 Structure of the FNN model for the range-based localization

Layers	FNN
1	Input
2	Dense (512, ReLU)
3	Dropout
4	Dense (256, ReLU)
5	Dense (64, ReLU)
6	Dense (16, ReLU)
7	Output (2, linear)

Abbreviations: FNN, feed-forward neural network; ReLU, rectified linear unit.

regularization strategy. The ReLU activation function is used in the hidden layers to reduce the likelihood of vanishing gradient while the linear activation function is used in the output layer to produce the real-value target's coordinate.

To generate the N_m ensemble models, random initial weights are used in each round of the training. The models are trained to minimize the mean-squared error cost function, where the adaptive moment estimation (ADAM) is selected as the optimization algorithm since it only requires small tuning of the hyper-parameters [49]. To avoid overtraining, early stopping is used. After the training, we have N_m trained models, which are used to generate N_m outputs for the location estimation unit.

6.1.2 | Effects of the training sample size

As the model's performance depends heavily on the number of training samples, we first vary the number of training samples to see its effects on the RMSE of the estimated target's location. A model is trained on a various number of training samples, varying from 100,000 to 1,000,000 samples. After the model is trained, the RMSE of the estimated target's location is computed from a separated 100,000 testing samples. The RMSE at a different number of training samples is plotted in Figure 16.

Here, the number of anchor N is set to 3, with $\gamma = 4$, $\sigma_r = 1 \text{ dB}$, and $\sigma_a = 5^\circ$. We denote NRNA as the hybrid RSS/AOA methods, which are based on N RSS and N AOA measurements, and LS, WLS, and CRLB as the least-squares method, the weighted least-squares method, and the CRLB given in Ref. [30], respectively.

Results show that the larger the training samples, the lower the error. For this problem, at least 300,000 training samples are required for the FNN to achieve the lower RMSE compared with the hybrid RSS/AOA methods. In practice, data augmentation techniques, such as noise injection, can be used to reduce the burden of collecting that large amount of samples.

6.1.3 | Effects of the number of ensemble models (N_m)

After the base model is sought, it is trained using 1,000,000 training samples with random initial weights N_m times, thus creating N_m ensemble models of the same structure but different sets of weights. By using the model-averaging ensemble method presented in Equation (51), the final target's location is obtained. Figure 17 shows the RMSE of the estimated target's location obtained from the ensemble method using a different number of ensemble models N_m .

As shown in the plot, the minimum RMSE obtained from a single model is around 1.41 m. Adding a second model to the computation reduces the RMSE to around 1.38 m. Results show that increasing N_m lowers the RMSE until a saturated point is reached at around 1.34 m at $N_m \approx 9$ –10 models.

FIGURE 16 RMSE versus number of training samples. CRLB, Cramer–Rao lower bound; FNN, feed-forward neural network; LS, least square; RMSE, root-mean-square error; WLS, weighted least squares

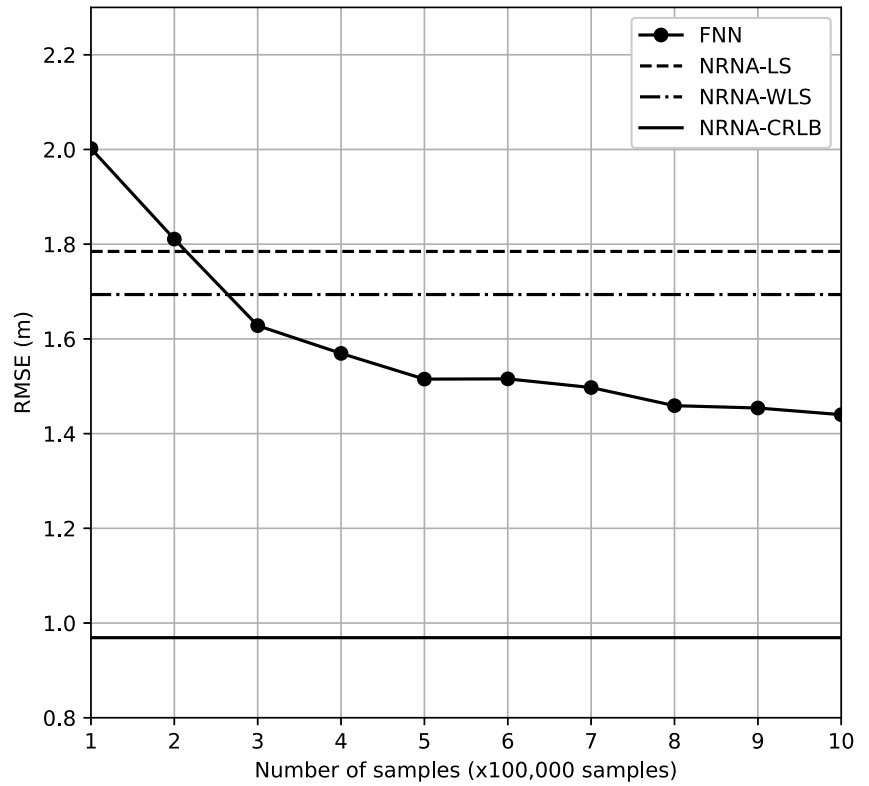
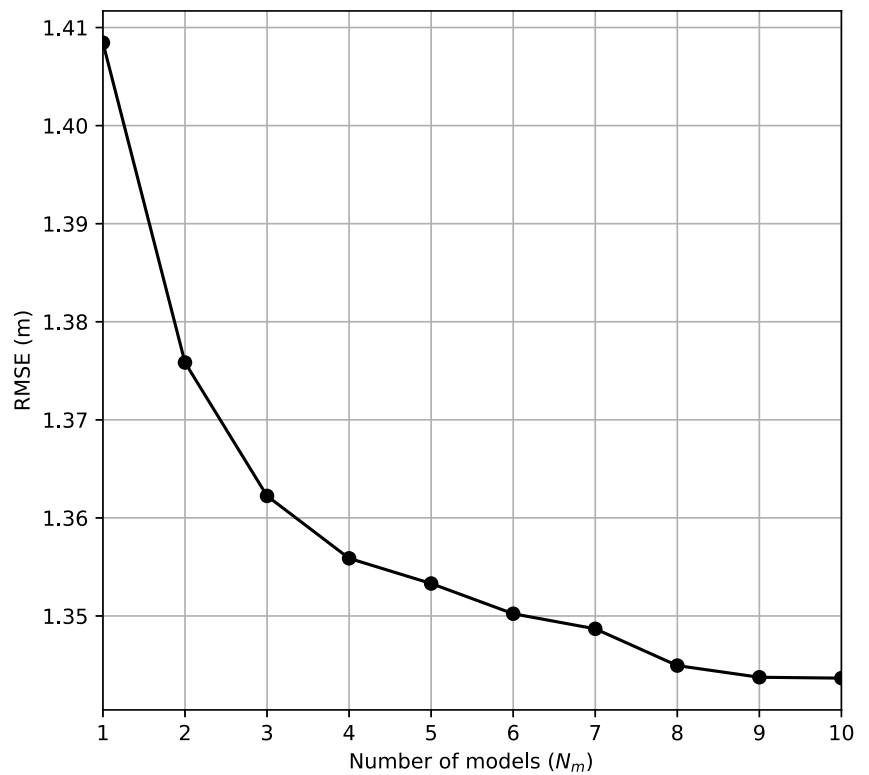


FIGURE 17 Root-mean-square error (RMSE) versus number of ensemble models



6.1.4 | Effects of the number of anchors (N)

Using $N_m = 10$, we next vary the number of anchors N to see its effects on the RMSE of the estimated target's location.

Figure 18 shows the RMSE when N is varied from 3 to 7. The solid lines represent the use of N RSS and N AOA (NRNA) input values, whereas the dash lines represent the use of N RSS and 1 AOA (NR1A) input values. The RMSE obtained from

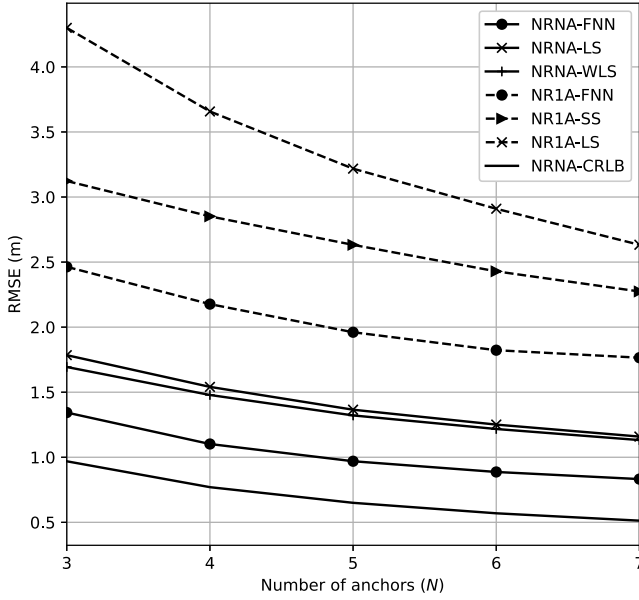


FIGURE 18 RMSE versus number anchors N . CRLB, Cramer–Rao lower bound; FNN, feed-forward neural network; LS, least square; RMSE, root-mean-square error; SS, subspace; WLS, weighted least squares

the proposed FNN, LS, WLS, and SS methods are compared with the CRLB given in Ref. [30].

Results show that the RMSE decreases as N increases for both NRNA and NR1A inputs. This is due to the fact that more useful information is added to the calculation for the target's location. From the plot, the methods using NRNA input are able to achieve lower RMSE compared with the methods using NR1A input, with the FNN providing the lowest RMSE in its respective NRNA and NR1A input types.

6.2 | Experimentation on the range-free dataset

For the range-free dataset, the training samples are formatted as shown in Equation (39). The input parameters consist of the observed RSS, observed CSI, whereas the output is the reference location ID. We evaluate the performance of the proposed DNN localization in two types of environments, that is, a crowded closed room and a corridor. The experiment settings are as follows.

6.2.1 | Experiment setting

In this experiment, a single anchor is used to locate a single target. The anchor is a Wi-Fi AP equipped with one antenna ($N_t = 1$) while the target is a laptop equipped with three antennas ($N_r = 3$). As a result, the information extracted from a packet reading consists of three RSS values and 3×30 complex CSI values, or $3 \times (30 \text{ real CSI magnitudes and } 30 \text{ real CSI phases})$. The training and testing locations for the closed room

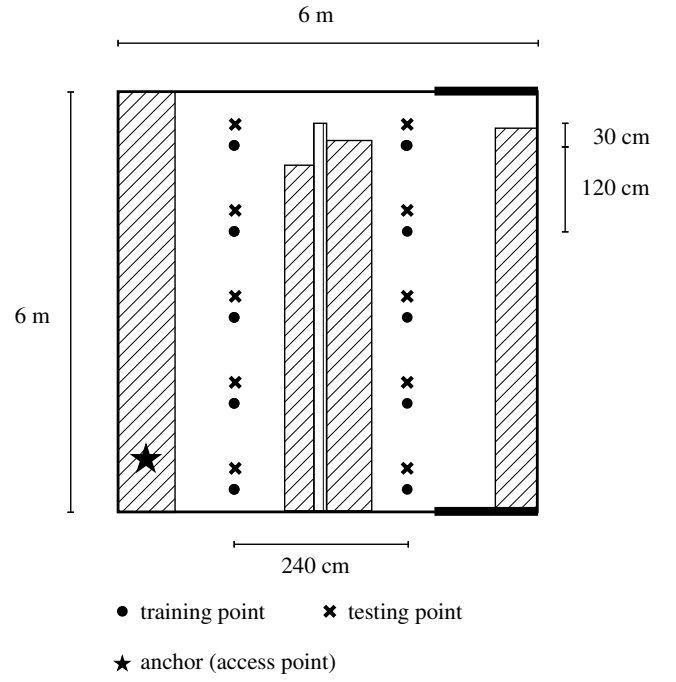


FIGURE 19 Training and testing locations for the closed room setting

and corridor settings are selected so that they do not coincide with each other, as shown in Figures 19 and 20, respectively.

The base model considered in this classification task includes the FNN, 1D-CNN, 2D-CNN, and LSTM. To train the base models, the same CSI values are used as the input for all base models; however, the values are formatted differently, as shown in Table 3. For the FNN base model, the CSI values are flattened into a 180-feature vector, whereas for the 1D-CNN and LSTM base models, the CSI values are considered as a series of data points in the frequency domain over the 30 subcarrier groups. For the 2D-CNN base model, the CSI values are formatted into a rectangular image.

The structure for the base models in closed room and corridor settings are shown in Tables 4 and 5, respectively. Note that the models presented here are merely a proof of concept to show that there exists a DNN model that performs better than the existing methods. A more advanced DNN architecture may provide higher localization accuracy, but it is not our main focus.

For the closed room setting, the FNN base model is a simple 1 hidden layer network as this is enough to achieve the lower RMSE than the existing fingerprinting methods. The ReLU activation function is used in the hidden layer to minimize the vanishing gradient, while the softmax activation is used in the output layer to generate categorical outputs. The 1D-CNN and 2D-CNN base models are similar in that they use a convolutional layer to extract the input features, followed by a pooling layer to down-sampling the extracted features. The difference is that the filters in the convolutional layer are 1D and 2D for the 1D-CNN and 2D-CNN, respectively. Lastly, a single LSTM layer is used in the LSTM base model. From our experiment, we found that more complex base

models are required for the corridor setting as there are not as many unique multipath components as in the closed room setting. Additional hidden layers and filters are added to the FNN and LSTM, and the 2D-CNN base models, whereas no adjustments are needed for the 1D-CNN base model.

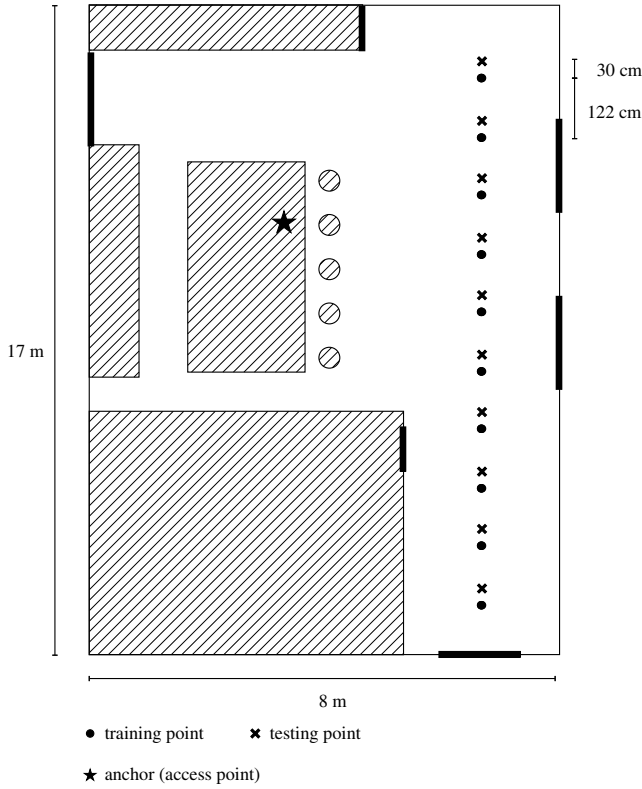


FIGURE 20 Training and testing locations for the corridor setting

TABLE 3 CSI input format

Deep neural network type	CSI input format
Feed-forward neural network	1 step \times 180 features
1D-CNN	30 frequency steps \times 6 features
2D-CNN	30 frequency steps \times 6 features image
Long short-term memory network	30 frequency steps \times 6 features

Abbreviations: CNN, convolutional neural network; CSI, channel state information.

TABLE 4 Structure of the base models for the closed room setting

Layers	Feed-forward neural network	1D-CNN	2D-CNN	LSTM
1	Input (180)	Input (30 \times 6)	Input (30 \times 6)	Input (30 \times 6)
2	Dense (16, ReLU)	Conv1D (filters = 16, kernel size = 9, ReLU)	Conv2D (filters = 20, kernel size = (13, 6), ReLU)	LSTM (40)
3	Output (10, Softmax)	GlobalMaxPooling1D	GlobalMaxPooling2D	Output (10, Softmax)
4	-	Output (10, Softmax)	Output (10, Softmax)	-

Abbreviations: CNN, convolutional neural network; LSTM, long short-term memory network; ReLU, rectified linear unit.

Here, the base models are trained to minimize the categorical cross-entropy cost function using the ADAM optimizer. Early stopping is also used to avoid overtraining. To train the models, we use a total sample size of 40,000 samples, with 4000 samples per reference location. The observed RSS and CSI values are first processed in the data pre-processing unit, where the RSS is filtered using a moving average window size of $W = 5$, and the CSI is calibrated using $l = 10$ first channel taps in the multipath filtering and sliding window size of $w = 3$ packets in the outlier filtering. Similar to the range-based dataset, $N_m = 10$ ensemble models are used. To evaluate the performance of the proposed DNN models, a separate 20,000 testing samples collected from the testing locations are used. The cumulative distribution function (CDF) of the distance error for the closed room and the corridor are plotted in Figures 21 and 22.

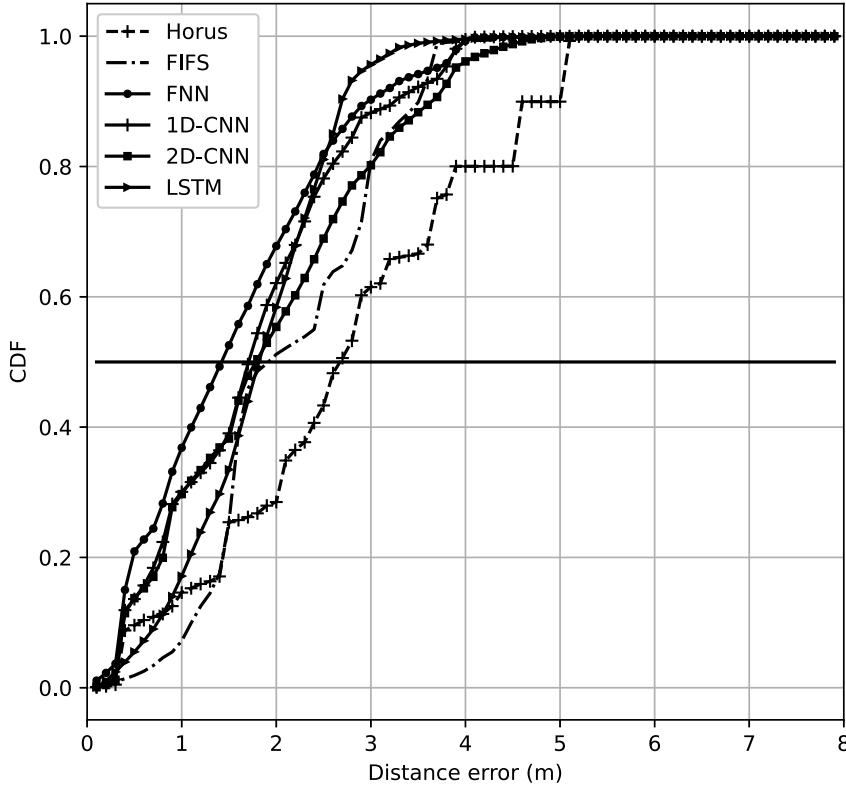
According to the two CDF plots, the maximum distance error is around 5 m for the closed room setting and is around 9 m for the corridor setting. The 50th percentile error for the proposed DNN models is all lower than Horus and FIFS for both settings. It is also worth mentioning that, in the closed room setting, Horus that uses the RSS measurement in fingerprinting performs worse than FIFS that uses the CSI measurement in terms of the 50th percentile error. However, in the corridor setting, Horus is able to achieve a lower distance error compared with FIFS. This means that the performance of the existing fingerprinting methods not only depends on the type of the input parameters but also on the selected features. One of the important reasons for the DNN approach to perform better than the existing methods is that the features are automatically learnt, not manually hand-picked.

The overall RMSE of the estimated target's location obtained from the ensemble method for both closed room and corridor settings are listed in Table 6. For the closed room setting, Horus has the highest RMSE of 3.08 m, followed by FIFS that has a RMSE of 2.34 m. The proposed DNN models are able to achieve the lower RMSE compared with the two existing methods at 1.86, 2.02, 2.23, and 1.95 m for the FNN, 1D-CNN, 2D-CNN, and LSTM, respectively. For the corridor setting, the RMSE shares a similar trend with the closed room setting. Horus has the highest RMSE of 4.12 m, followed by FIFS that has the RMSE of 3.50 m. All proposed DNN models offer the lower RMSE at 3.37, 3.26, 3.18, and 3.40 m for the FNN, 1D-CNN, 2D-CNN and LSTM, respectively.

TABLE 5 Structure of the base models for the corridor setting

Layers	FNN	1D-CNN	2D-CNN	LSTM
1	Input (180)	Input (30×6)	Input (30×6)	Input (30×6)
2	Dense (6, ReLU)	Conv1D (filters = 16, kernel size = 9, ReLU)	Conv2D (filters = 32, kernel size = (13, 6), ReLU)	LSTM (44)
3	Dense (4, ReLU)	GlobalMaxPooling1D	GlobalMaxPooling2D	LSTM (16)
4	Dense (4, ReLU)	Output (10, Softmax)	Output (10, Softmax)	LSTM (12)
5	Dropout	-	-	Output (10, Softmax)
7	Dense (4, ReLU)	-	-	-
8	Output (10, Softmax)	-	-	-

Abbreviations: CNN, convolutional neural network; LSTM, long short-term memory network; ReLU, rectified linear unit.

**FIGURE 21** CDF of the distance error in the closed-room setting. CDF, cumulative distribution function; CNN, convolutional neural network; FIFS, Fine-grained Indoor Fingerprint System; FNN, feed-forward neural network; LSTM, long short-term memory network

We next investigate the effects of the ensemble method. The RMSE obtained from a single DNN model versus the RMSE obtained from an ensemble of $N_m = 10$ models are listed in Table 7 for the closed room setting and Table 8 for the corridor setting. In both settings, we observe a high variation of the RMSE obtained from the single-model method, where the difference between the minimum and the maximum RMSE obtained from the same DNN structure in different training sessions exceeds 1 m, and some of the single models perform even worse than Horus and FIFS. By using the ensemble method, we are able to obtain the RMSE that is lower than the existing methods as well as the average RMSE obtained from a single method for all types of DNN models. In the case where the single model with the highest RMSE is used, the difference between the RMSE obtained from the single model and the ensemble methods could be as

large as 1.47 m, as seen in the case of the FNN model in the corridor setting. Moreover, the RMSE obtained from the ensemble method for the FNN, 2D-CNN, and LSTM is even lower than the minimum RMSE obtained from their single-model counterparts.

7 | CONCLUSION

In this work, we investigate the performance of the DNN-based wireless indoor localization when the models are trained on the range-based and range-free datasets. Several types of DNN models are considered, including the FNN, 1D-CNN, 2D-CNN, and LSTM. A model ensemble technique is also proposed to improve the performance of the target localization.

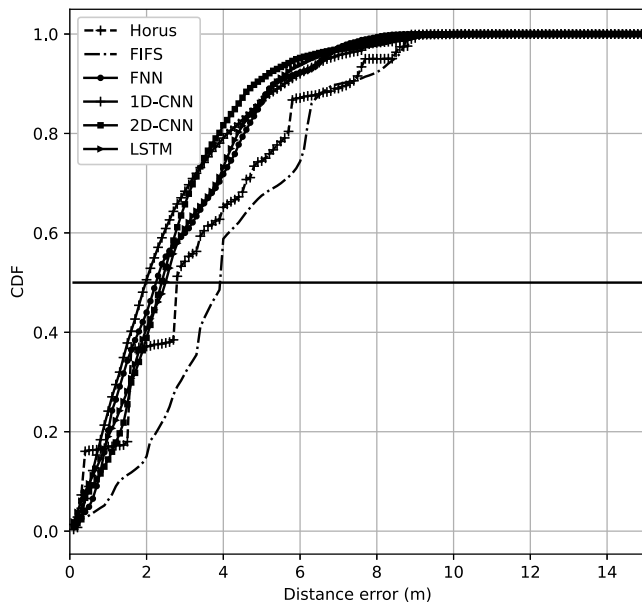


FIGURE 22 CDF of the distance error in the corridor setting. CDF, cumulative distribution function; CNN, convolutional neural network; FIFS, Fine-grained Indoor Fingerprint System; FNN, feed-forward neural network; LSTM, long short-term memory network

TABLE 6 Root-mean-square error (m) obtained from the ensemble method for the closed room and corridor settings

Methods	Closed room	Corridor
Horus	3.08	4.12
Fine-grained Indoor Fingerprint System	2.34	3.50
Feed-forward neural network	1.86	3.37
1D-CNN	2.02	3.26
2D-CNN	2.23	3.18
Long short-term memory network	1.95	3.40

Abbreviation: CNN, convolutional neural network.

TABLE 7 Root-mean-square error (m) comparison between a single-model method and an ensemble method in the closed-room setting

Methods	Single model			Ensemble
	Min	Max	Average	
Feed-forward neural network	1.97	2.53	2.14	1.86
1D-CNN	1.99	2.51	2.22	2.02
2D-CNN	2.25	2.58	2.41	2.23
Long short-term memory network	1.96	2.69	2.36	1.95

Abbreviation: CNN, convolutional neural network.

For the range-based dataset, we found that the proposed DNN model is able to achieve a lower RMSE compared with the existing hybrid RSS/AOA methods, given that enough training samples are provided. Fortunately, data augmentation can be easily applied to generate more training samples.

For the range-free dataset, we found that the proposed DNN models provide a lower RMSE compared with the

TABLE 8 Root-mean-square error (m) comparison between a single-model method and an ensemble method in a corridor setting

Methods	Single model			Ensemble
	Min	Max	Average	
Feed-forward neural network	3.45	4.84	3.97	3.37
1D-CNN	3.08	4.53	3.94	3.26
2D-CNN	3.35	3.92	3.71	3.18
Long short-term memory network	3.49	4.75	3.96	3.40

Abbreviation: CNN, convolutional neural network.

existing fingerprinting methods, that is, Horus and FIFS, in both closed room and corridor settings. This is because the proposed DNN models have the ability to automatically extract useful features, unlike the existing methods where the features are hand-picked. We also found that by using the ensemble technique, the obtained RMSE can be even lower than the minimum RMSE obtained from a single model for the FNN, 2D-CNN, and LSTM models.

As a result, the proposed DNN approach offers a better performance than the traditional methods. A DNN regression model can be applied to a system where range-based trilateration and triangulation are preferred, such as a large area with LOS, while a DNN classification model can be applied to a system where the range-free fingerprinting is preferred, such as a relatively smaller area with NLOS. In addition, by using the ensemble method, a more stable target estimation and higher localization accuracy can be achieved. Based on our experiment, we believe that the use of model ensemble could be considered as a good practice when employing deep learning in wireless localization.

Our future work includes several adjustments to the ensemble technique. In this work, we only consider combining the models of the same structure. By combining different structures or different types of DNN models, the localization accuracy might be improved. In addition, we currently combine the models randomly, adding criteria, such as applying a threshold on the training accuracy, for the model selection for the ensemble method should be further investigated.

ACKNOWLEDGEMENTS

We would like to thank the National e-Science Infrastructure Consortium, Thailand, for providing computing resources that have contributed to the research results reported within this paper. We also would like to thank the New South Wales (NSW) Defense Innovation Network (DIN) and the NSW State Government, Australia, for the financial support through the DIN Pilot Project Grant (Project ID: 888-006-985, Funding Years: 2019–2021).

CONFLICT OF INTEREST

All authors report no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Juthatip Wisanmongkol  <https://orcid.org/0000-0002-8101-8506>

REFERENCES

- Viani, F., et al.: Opportunistic exploitation of wireless infrastructures for homeland security. In: IEEE International Symposium on Antennas and Propagation (APSURSI), pp. 3062–3065. (2011)
- Amaresh, H.S., Rao, Y.G.A., Hallikar, R.S.: Real-time intruder detection system using sound localization and background subtraction. In: Texas Instruments India Educators' Conference (THIEC), pp. 131–137. (2014)
- Lee, C.K.M., et al.: A Bluetooth location-based indoor positioning system for asset tracking in warehouse. In: IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), pp. 1408–1412. (2019)
- Kao, C.H., et al.: A hybrid indoor positioning for asset tracking using Bluetooth Low Energy and Wi-Fi. In: IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), pp. 63–64. (2017)
- Rajendran, N., et al.: WATS-SN: a wireless asset tracking system using sensor networks. In: IEEE International Conference on Personal Wireless Communications, 2005. ICPWC, pp. 237–243. (2005)
- Silventoinen, M.I., Rantalainen, T.: Mobile station emergency locating in GSM. In: IEEE International Conference on Personal Wireless Communications Proceedings and Exhibition, Future Access, pp. 232–238. (1996)
- Swales, S.C., Maloney, J.E., Stevenson, J.O.: Locating mobile phones and the US wireless E-911 mandate. In: IEE Colloquium on Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications (Ref. No. 1999/046), pp. 2/1–2/6. (1999)
- Logan, L., Davids, C., Davids, C.: Determining the indoor location of an emergency caller in a multi-story building. In: IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), pp. 1–6. (2020)
- Ito, T., Anzai, D., Wang, J.: Novel joint TOA/RSSI-based WCE location tracking method without prior knowledge of biological human body tissues. In: 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 6993–6996. (2014)
- Ara, P., Heimlich, M., Dutkiewicz, E.: Investigation of radar approach for localization of gastro intestinal endoscopic capsule. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 99–104. (2014)
- Nafchi, A.R., Goh, S.T., Zekavat, S.A.R.: High performance DOA/TOA-based endoscopy capsule localization and tracking via 2D circular arrays and inertial measurement unit. In: IEEE International Conference on Wireless for Space and Extreme Environments, pp. 1–6. (2013)
- Rahimi, H., et al.: Indoor geo-fencing and access control for wireless networks. In: IEEE Symposium on Computational Intelligence in Cyber Security (CICS), pp. 1–8. (2013)
- Avdyushkin, M., Rahman, M.: Secure location validation with Wi-Fi geofencing and NFC. IEEE Trustcom/BigDataSE/ISPA 1, 890–896. (2015)
- Spachos, P., Plataniotis, K.N.: BLE beacons for indoor positioning at an interactive IoT-based smart museum. IEEE Syst. J. 14(3), 3483–3493. (2020)
- Polo, A., et al.: Semantic wireless localization enabling advanced services in museums. In: The 8th European Conference on Antennas and Propagation (EuCAP 2014), pp. 443–446. (2014)
- Kovavisaruch, L., et al.: Museums pool: a mobile application for museum network. In: Portland International Conference on Management of Engineering and Technology (PICMET), pp. 1230–1235. (2015)
- Gu, Y., Ren, F.: Energy-efficient indoor localization of smart hand-held devices using Bluetooth. IEEE Access. 3, 1450–1461. (2015)
- Dinh, T.M.T., Duong, N.S., Sandrasegaran, K.: Smartphone-based indoor positioning using BLE iBeacon and reliable lightweight fingerprint map. IEEE Sensor. J. 20, 10283–10294. (2020)
- Mahfouz, M.R., et al.: Investigation of high-accuracy indoor 3-D positioning using UWB technology. IEEE Trans. Microw. Theor. Tech. 56, 1316–1330. (2008)
- Alsindi, N.A., Alavi, B., Pahlavan, K.: Measurement and modeling of ultrawideband TOA-based ranging in indoor multipath environments. IEEE Trans. Veh. Technol. 58, 1046–1058. (2009)
- Meissner, P., Leitinger, E., Witrisal, K.: UWB for robust indoor tracking: weighting of multipath components for efficient estimation. IEEE Wirel. Commun. Lett. 3, 501–504. (2014)
- Barbi, M., et al.: UWB RSS-based localization for capsule endoscopy using a multilayer phantom and in vivo measurements. IEEE Trans. Antenn. Propag. 67, 5035–5043. (2019)
- Cheung, K., Intille, S., Larson, K.: An inexpensive Bluetooth-based indoor positioning hack. In: International Conference of Ubiquitous Computing (UbiComp2006), pp. 1–2. (2006)
- Liu, F., et al.: Survey on WiFi-based indoor positioning techniques. IET Commun. 14, 1372–1383. (2020)
- Wang, Y.: Linear least squares localization in sensor networks. EURASIP J. Wirel. Commun. Netw. 1, 51. (2015)
- Sayed, A.H., Tarighat, A., Khajehnouri, N.: Network-based wireless location: challenges faced in developing techniques for accurate wireless location information. IEEE Signal Process. Mag. 22(4), 24–40. (2005)
- Tomic, S., et al.: A closed-form solution for RSS/AoA target localization by spherical coordinates conversion. IEEE Wirel. Commun. Lett. 5(6), 680–683. (2016)
- Ding, W., Chang, S., Li, J.: A novel weighted localization method in wireless sensor networks based on hybrid RSS/AoA measurements. IEEE Access. 9, 150677–150685. (2021)
- Nguyen, N.M., et al.: Performance evaluation of non-GPS based localization techniques under shadowing effects. Sensors. 19, 2633. (2019)
- Le, A.T., et al.: Unbalanced hybrid AOA/RSS localization for simplified wireless sensor networks. Sensors. 1, 1–14. (2020)
- Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000. Tel Aviv, Israel, pp. 775–784. (2000)
- Youssef, M., Agrawala, A.: The Horus WLAN location determination system. In: Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services. MobiSys '05. New York, NY, USA: Association for Computing Machinery, pp. 205–218. (2005)
- Yang, Z., Zhou, Z., Liu, Y.: From RSSI to CSI: indoor localization via channel response. ACM Comput. Surv. 46(2), 1–32. (2013)
- Xiao, J., et al.: FIFS: fine-grained indoor fingerprinting system. In: 21st International Conference on Computer Communications and Networks (ICCCN), pp. 1–7. (2012)
- Wang, X., Wang, X., Mao, S.: CiFi: deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–6. (2017)
- Hsieh, C.H., Chen, J.Y., Nien, B.H.: Deep learning-based indoor localization using received signal strength and channel state information. IEEE Access. 7, 33256–33267. (2019)
- Bencharif, L., Ouameur, M.A., Massicotte, D.: Long short-term memory for indoor localization using WI-FI received signal strength and channel state information. In: 2021 IEEE 4th 5G World Forum (5GWF), pp. 230–235. (2021)
- Njima, W., et al.: Indoor localization using data augmentation via selective generative adversarial networks. IEEE Access. 9, 98337–98347. (2021)
- Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Network. 2(5), 359–366. (1989)
- Chen, C., et al.: A survey on deep learning for localization and mapping: towards the age of spatial machine intelligence. CoRR. 12567, 1–26. (2020). abs/2006
- Kharb, K., Sharma, B., Trilok, C.A.: Reliable and congestion control protocols for wireless sensor networks. Int. J. Eng. Sci. Innovat. Technol. 6(1), 68–78. (2016)
- Sharma, B., Aseri, T.C.: A hybrid and dynamic reliable transport protocol for wireless sensor networks. Comput. Electr. Eng. 48, 298–311. (2015)
- Chand, T., Sharma, B.: HRCCTP: a hybrid reliable and congestion control transport protocol for wireless sensor networks. In: 2015 IEEE SENSORS, pp. 1–4. (2015)

44. So, H.C., Chan, F.K.W.: A generalized subspace approach for mobile positioning with time-of-arrival measurements. *IEEE Trans. Signal Process.* 55, 5103–5107. (2007)
45. Halperin, D., et al.: Tool release: gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR* 41(1), 53. (2011)
46. IEEE standard for information technology—local and metropolitan area networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: enhancements for higher throughput, *IEEE Std 80211n-2009*, pp. 1–565. (2009)
47. Sen, S., et al.: You are facing the Mona Lisa: spot localization using PHY layer information. In: *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. MobiSys '12*. New York, NY, USA: Association for Computing Machinery, pp. 183–196. (2012)

48. Wu, C., et al.: PhaseU: real-time LOS identification with WiFi. In: *IEEE Conference on Computer Communications (INFOCOM)*, pp. 2038–2046. (2015)
49. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* (2015)

How to cite this article: Wisanmongkol, J., et al.: An ensemble approach to deep-learning-based wireless indoor localization. *IET Wirel. Sens. Syst.* 12(2), 33–55. (2022). <https://doi.org/10.1049/wss2.12035>

APPENDIX

To derive the CRLB for the CSI-based localization, we first consider a localization system as shown in Figure A1. We assume that the anchor is centred at $\mathbf{a} = [a_x, a_y]^\top$ and the target is centred at $\mathbf{b} = [b_x, b_y]^\top$. The anchor is equipped with N_t antennas, where the i th element is located at $\mathbf{r}_{t,i} = [r_{t,i} \cos(\phi_{t,i}), r_{t,i} \sin(\phi_{t,i})]^\top$ for $i = 1, \dots, N_t$. Similarly, the target is equipped with N_r antennas, where the i th element is located at $\mathbf{r}_{r,i} = [r_{r,i} \cos(\phi_{r,i}), r_{r,i} \sin(\phi_{r,i})]^\top$ for $i = 1, \dots, N_r$. A total of $K + 1$ paths exist between the anchor and the target. The direction for the DOD and DOA of the k th path are characterized by the unit vector $\boldsymbol{\Omega}_{t,k} = [\cos(\theta_{t,k}), \sin(\theta_{t,k})]^\top$ and $\boldsymbol{\Omega}_{r,k} = -[\cos(\theta_{r,k}), \sin(\theta_{r,k})]^\top$, respectively.

For a transmission symbol g , the $N_r \times N_t$ CFR for the n th subcarrier is given by

$$\mathbf{H}_n = \sum_{k=0}^K h_k \exp\left\{-j2\pi n \tau_k\right\} \mathbf{a}_r(\boldsymbol{\Omega}_{r,k}) \mathbf{a}_t^\top(\boldsymbol{\Omega}_{t,k}), \quad (\text{A.1})$$

where h_k and τ_k are the complex channel gain and the TOA for the k th path, N_{sc} is the number of subcarriers, T_s is the sampling period, $\mathbf{a}_r(\boldsymbol{\Omega}_{r,k})$ and $\mathbf{a}_t(\boldsymbol{\Omega}_{t,k})$ are the target's and the anchor's antenna steering vectors such that

$$\mathbf{a}_r(\boldsymbol{\Omega}_{r,k}) = \left[e^{j2\pi \frac{\boldsymbol{\Omega}_{r,k} \cdot \mathbf{r}_{r,1}}{\lambda_c}} \quad \dots \quad e^{j2\pi \frac{\boldsymbol{\Omega}_{r,k} \cdot \mathbf{r}_{r,N_r}}{\lambda_c}} \right]^\top, \quad (\text{A.2})$$

and

$$\mathbf{a}_t(\boldsymbol{\Omega}_{t,k}) = \left[e^{j2\pi \frac{\boldsymbol{\Omega}_{t,k} \cdot \mathbf{r}_{t,1}}{\lambda_c}} \quad \dots \quad e^{j2\pi \frac{\boldsymbol{\Omega}_{t,k} \cdot \mathbf{r}_{t,N_t}}{\lambda_c}} \right]^\top, \quad (\text{A.3})$$

where λ_c is the wavelength of the carrier frequency, $\boldsymbol{\Omega}_{r,k} \cdot \mathbf{r}_{r,i} = -r_{r,i} \cos(\theta_{r,k} - \phi_{r,i})$ and $\boldsymbol{\Omega}_{t,k} \cdot \mathbf{r}_{t,i} = r_{t,i} \cos(\theta_{t,k} - \phi_{t,i})$, respectively.

The received signal on the n th subcarrier, \mathbf{Y}_n , is defined as

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{x}_n + \mathbf{w}_n, \quad (\text{A.4})$$

where \mathbf{x}_n is a $N_t \times 1$ vector of transmitted signal, and \mathbf{w}_n is an additive white Gaussian noise with zero mean and variance of $\frac{\sigma_w^2}{2}$ per real dimension.

We define ξ as a vector of unknown parameters of all $K + 1$ paths, that is,

$$\xi = [\tau_0, \theta_{r,0}, \theta_{t,0}, h_{R,0}, h_{I,0}, \dots, \tau_K, \theta_{r,K}, \theta_{t,K}, h_{R,K}, h_{I,K}], \quad (\text{A.5})$$

where $h_{R,k}$ and $h_{I,k}$ are the real and imaginary parts of h_k , respectively. For a complex Gaussian data vector, the Fisher information matrix $\mathbf{I}(\xi)$ can be computed from

$$\begin{aligned} [\mathbf{I}(\xi)]_{ij} = & \sum_{n=1}^{N_{sc}} \text{tr} \left[\mathbf{C}_{\mathbf{Y}_n}^{-1}(\xi) \frac{\partial \mathbf{C}_{\mathbf{Y}_n}(\xi)}{\partial \xi_i} \mathbf{C}_{\mathbf{Y}_n}^{-1}(\xi) \frac{\partial \mathbf{C}_{\mathbf{Y}_n}(\xi)}{\partial \xi_j} \right] \\ & + 2\text{Re} \left[\frac{\partial \mu_n^H(\xi)}{\partial \xi_i} \mathbf{C}_{\mathbf{Y}_n}^{-1}(\xi) \frac{\partial \mu_n(\xi)}{\partial \xi_j} \right], \end{aligned} \quad (\text{A.6})$$

where $\mathbf{C}_{\mathbf{Y}_n}(\xi) = \sigma_w^2 \mathbf{I}$ is the covariance matrix of \mathbf{Y}_n , and $\mu_n = \mathbf{H}_n \mathbf{x}_n$. Therefore, Equation (A.6) can be written as

$$[\mathbf{I}(\xi)]_{ij} = \sum_{n=1}^{N_{sc}} \frac{2}{\sigma_w^2} \text{Re} \left[\frac{\partial \mu_n^H(\xi)}{\partial \xi_i} \frac{\partial \mu_n(\xi)}{\partial \xi_j} \right]. \quad (\text{A.7})$$

The values for the components of the FIM are listed below.

$$\begin{aligned} \frac{\partial \mu_n(\xi)}{\partial \tau_i} = & \left(\frac{-j2\pi n}{N_{sc} T_s} \right) h_i \exp \left\{ \frac{-j2\pi n \tau_i}{N_{sc} T_s} \right\} \\ & \times \mathbf{a}_r(\boldsymbol{\Omega}_{r,i}) \mathbf{a}_t^\top(\boldsymbol{\Omega}_{t,i}) \mathbf{x}_n, \end{aligned} \quad (\text{A.8})$$

$$\begin{aligned} \frac{\partial \mu_n(\xi)}{\partial \theta_{r,i}} = & h_i \exp \left\{ \frac{-j2\pi n \tau_i}{N_{sc} T_s} \right\} \\ & \times \mathbf{D}_{r,i} \mathbf{a}_r(\boldsymbol{\Omega}_{r,i}) \mathbf{a}_t^\top(\boldsymbol{\Omega}_{t,i}) \mathbf{x}_n, \end{aligned} \quad (\text{A.9})$$

$$\begin{aligned} \frac{\partial \mu_n(\xi)}{\partial \theta_{t,i}} = & h_i \exp \left\{ \frac{-j2\pi n \tau_i}{N_{sc} T_s} \right\} \\ & \times \mathbf{a}_r(\boldsymbol{\Omega}_{r,i}) (\mathbf{D}_{t,i} \mathbf{a}_t)^\top(\boldsymbol{\Omega}_{t,i}) \mathbf{x}_n, \end{aligned} \quad (\text{A.10})$$

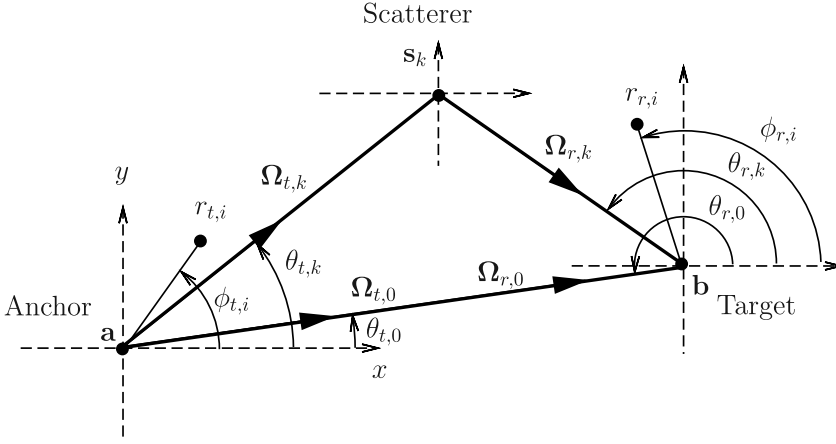


FIGURE A1 Channel-state-information-based localization system model

$$\frac{\partial \mu_n(\xi)}{\partial b_{R,i}} = \exp\left\{\frac{-j2\pi n \tau_i}{N_{sc} T_s}\right\} \times \mathbf{a}_r(\boldsymbol{\Omega}_{r,i}) \mathbf{a}_t^\top(\boldsymbol{\Omega}_{t,i}) \mathbf{x}_n, \quad (\text{A.11})$$

$$\frac{\partial \mu_n(\xi)}{\partial b_{L,i}} = j \exp\left\{\frac{-j2\pi n \tau_i}{N_{sc} T_s}\right\} \times \mathbf{a}_r(\boldsymbol{\Omega}_{r,i}) \mathbf{a}_t^\top(\boldsymbol{\Omega}_{t,i}) \mathbf{x}_n, \quad (\text{A.12})$$

where

$$\mathbf{D}_{r,i} = \text{diag}\left\{\frac{-j2\pi}{\lambda_n} r_{r,1} \sin(\theta_{r,i} - \phi_{r,1}), \dots, \frac{-j2\pi}{\lambda_n} r_{r,N_r} \sin(\theta_{r,i} - \phi_{r,N_r})\right\}, \quad (\text{A.13})$$

$$\mathbf{D}_{t,i} = \text{diag}\left\{\frac{j2\pi}{\lambda_n} r_{t,1} \sin(\theta_{t,i} - \phi_{t,1}), \dots, \frac{j2\pi}{\lambda_n} r_{t,N_t} \sin(\theta_{t,i} - \phi_{t,N_t})\right\}. \quad (\text{A.14})$$

To relate the FIM of the channel parameters with target positioning, a reparametrization is performed by introducing a vector of unknown positions

$$\mathbf{P} = [b_x, b_y, s_{1,x}, s_{1,y}, \dots, s_{K,x}, s_{K,y}]^\top, \quad (\text{A.15})$$

where b_x and b_y are the coordinates of the target, and $s_{k,x}$ and $s_{k,y}$ are the coordinates of the scatterer on the k th path. The relationships between these position parameters \mathbf{P} and the channel parameters ξ are described by

$$\tau_0 = \frac{\|\mathbf{b} - \mathbf{a}\|}{c}, \quad (\text{A.16})$$

$$\theta_{t,0} = \sin^{-1}\left(\frac{b_y - a_y}{\|\mathbf{b} - \mathbf{a}\|}\right) = \cos^{-1}\left(\frac{b_x - a_x}{\|\mathbf{b} - \mathbf{a}\|}\right), \quad (\text{A.17})$$

$$\theta_{r,0} = \sin^{-1}\left(\frac{a_y - b_y}{\|\mathbf{a} - \mathbf{b}\|}\right) = \cos^{-1}\left(\frac{a_x - b_x}{\|\mathbf{a} - \mathbf{b}\|}\right), \quad (\text{A.18})$$

$$\tau_k = \frac{\|\mathbf{b} - \mathbf{s}_k\| + \|\mathbf{s}_k - \mathbf{a}\|}{c}, \quad (\text{A.19})$$

$$\theta_{t,k} = \sin^{-1}\left(\frac{s_{k,y} - a_y}{\|\mathbf{s}_k - \mathbf{a}\|}\right) = \cos^{-1}\left(\frac{s_{k,x} - a_x}{\|\mathbf{s}_k - \mathbf{a}\|}\right), \quad (\text{A.20})$$

$$\theta_{r,k} = \sin^{-1}\left(\frac{s_{k,y} - b_y}{\|\mathbf{s}_k - \mathbf{b}\|}\right) = \cos^{-1}\left(\frac{s_{k,x} - b_x}{\|\mathbf{s}_k - \mathbf{b}\|}\right). \quad (\text{A.21})$$

We define a Jacobian matrix \mathbf{J} as

$$\mathbf{J} = \frac{\partial \xi}{\partial \mathbf{P}}, \quad (\text{A.22})$$

where the elements for the LOS path can be computed as

$$\frac{\partial \tau_0}{\partial b_x} = \frac{1}{c} \cos(\theta_{t,0}), \quad (\text{A.23})$$

$$\frac{\partial \tau_0}{\partial b_y} = \frac{1}{c} \sin(\theta_{t,0}), \quad (\text{A.24})$$

$$\frac{\partial \theta_{t,0}}{\partial b_x} = \frac{-1}{\|\mathbf{b} - \mathbf{a}\|} \sin(\theta_{t,0}), \quad (\text{A.25})$$

$$\frac{\partial \theta_{t,0}}{\partial b_y} = \frac{1}{\|\mathbf{b} - \mathbf{a}\|} \cos(\theta_{t,0}), \quad (\text{A.26})$$

$$\frac{\partial \theta_{r,0}}{\partial b_x} = \frac{1}{\|\mathbf{a} - \mathbf{b}\|} \sin(\theta_{r,0}), \quad (\text{A.27})$$

$$\frac{\partial \theta_{r,0}}{\partial b_y} = \frac{-1}{\|\mathbf{a} - \mathbf{b}\|} \cos(\theta_{r,0}), \quad (\text{A.28})$$

the elements for the NLOS paths are

$$\frac{\partial \tau_k}{\partial b_x} = \frac{-1}{c} \cos(\theta_{r,k}), \quad (\text{A.29})$$

$$\frac{\partial \tau_k}{\partial b_y} = \frac{-1}{c} \sin(\theta_{r,k}), \quad (\text{A.30})$$

$$\frac{\partial \tau_k}{\partial s_{k,x}} = \frac{1}{c} \cos(\theta_{t,k}) + \frac{1}{c} \cos(\theta_{r,k}), \quad (\text{A.31})$$

$$\frac{\partial \tau_k}{\partial s_{k,y}} = \frac{1}{c} \sin(\theta_{t,k}) + \frac{1}{c} \sin(\theta_{r,k}), \quad (\text{A.32})$$

$$\frac{\partial \theta_{t,k}}{\partial s_{k,x}} = \frac{-1}{\|\mathbf{s}_k - \mathbf{a}\|} \sin(\theta_{t,k}), \quad (\text{A.33})$$

$$\frac{\partial \theta_{t,k}}{\partial s_{k,y}} = \frac{1}{\|\mathbf{s}_k - \mathbf{a}\|} \cos(\theta_{t,k}), \quad (\text{A.34})$$

$$\frac{\partial \theta_{r,k}}{\partial b_x} = \frac{1}{\|\mathbf{s}_k - \mathbf{b}\|} \sin(\theta_{r,k}), \quad (\text{A.35})$$

$$\frac{\partial \theta_{r,k}}{\partial b_y} = \frac{-1}{\|\mathbf{s}_k - \mathbf{b}\|} \cos(\theta_{r,k}), \quad (\text{A.36})$$

$$\frac{\partial \theta_{r,k}}{\partial s_{k,x}} = \frac{-1}{\|\mathbf{s}_k - \mathbf{b}\|} \sin(\theta_{r,k}), \quad (\text{A.37})$$

$$\frac{\partial \theta_{r,k}}{\partial s_{k,y}} = \frac{1}{\|\mathbf{s}_k - \mathbf{b}\|} \cos(\theta_{r,k}), \quad (\text{A.38})$$

and the other components are all zero. The FIM for position parameters can be written as

$$\mathbf{I}(\mathbf{P}) = \mathbf{J}^T \mathbf{I}(\xi) \mathbf{J}. \quad (\text{A.39})$$

The CRLB for the target localization error is then

$$\text{CRLB} = [\mathbf{I}^{-1}(\mathbf{P})]_{1,1} + [\mathbf{I}^{-1}(\mathbf{P})]_{2,2}. \quad (\text{A.40})$$