



EA-CNN: A smart indoor 3D positioning scheme based on Wi-Fi fingerprinting and deep learning

Atefe Alitaleshi, Hamid Jazayeriy*, Javad Kazemitabar

Faculty of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran



ARTICLE INFO

Keywords:

Extreme learning machine
Indoor positioning
Data augmentation
Convolutional neural network
Wi-Fi fingerprinting

ABSTRACT

Accurate indoor location information in multi-building/floor environments is essential for establishing many indoor location-based services (LBS). Wi-Fi fingerprinting with received signal strength indicators (RSSI) has become one of the most practical techniques to localize users indoors. However, the fluctuation of wireless signals caused by fading, the multipath effect, and device heterogeneity leads to considerable variations in RSSIs, which poses a challenge to accurate localization. This paper proposes an indoor positioning method, which is constructed based on a convolutional neural network (CNN) framework. Specifically, a novel model by combining an extreme learning machine autoencoder (ELM-AE) with a two-dimensional CNN is proposed. The ELM-AE extracts critical features by reducing input dimensions, while the CNN is trained to effectively achieve significant performance in the positioning phase. To increase positioning accuracy and deal with data shortages, a data augmentation strategy by frequently adding noisy data to the original fingerprint map is devised. The statistical properties namely, the RSS values of each MAC address, are utilized to adjust input noise. The performance of the proposed system is evaluated on two Tampere and UJIIndoorLoc datasets. The experimental results show that EA-CNN achieves better performance than CNN by decreasing average positioning error up to 40.95% and 43.74% in Tampere and UJIIndoorLoc datasets, respectively. Compared to state-of-the-art deep learning-based methods, the positioning performance improves up to 68.36% in Tampere and 67.56% in the UJIIndoorLoc dataset by exploiting only 25% of the training samples.

Compared to several state-of-the-art deep learning models, EA-CNN achieves higher accuracy in both positioning and floor estimation.

1. Introduction

With the growing development of the Internet of Things (IoT), the application of location-based services (LBS) has attracted wide attention from both industry and academic research (Uphaus et al., 2021). Since people spend most of their time indoors, like multi-story shopping malls, hospitals, museums, and airports, the request for indoor location-based services is in high demand (Basiri et al., 2017; Min et al., 2021). Nowadays, smartphones are ubiquitous and carried almost everywhere, which makes the deployment of indoor location-based services even more tangible. Accurate location determination plays a crucial role in quality-of-service delivery. The global positioning system (GPS) is widely used for determining the location in open/outdoor areas. However, the lack of line of sight (LOS) and the effect of multipath fading and shadowing on signals make it limited to utilize indoor positioning (Sithole and Zlatanova, 2016). Alternatives technologies such as Wireless Fidelity (Wi-Fi), Bluetooth, radio frequency identification (RFID), ultra-wideband (UWB), etc., have been proposed for indoor positioning. Among these technologies, Wi-Fi is cost-effective

and hence the most popular and feasible technology (Spachos and Plataniotis, 2020; Yao and Hsia, 2018; Dumbgen et al., 2019; Xia et al., 2017).

Wireless signal-based positioning methods are generally classified into two categories: geometric and fingerprinting methods. In geometric-based methods, the location is estimated according to various measurement parameters, including the time of arrival (ToA), time of flight (ToF), and angle of arrival (AoA). Although geometric-based methods are well established, non-line-of-sight effects, multipath problems, and the need for specialized antennae along with precise synchronization requirements make them less effective for practical indoor positioning. Instead, Wi-Fi fingerprinting is widely used for indoor positioning due to its ubiquity, cost-effectiveness, and no need for additional infrastructure (Mendoza-Silva et al., 2019; Liu et al., 2020).

Fingerprinting techniques exploit received signal strength indication (RSSI) or channel state information (CSI) to predict the target location (Geok et al., 2021). In the CSI-based positioning systems, information on a communication link, including rank indication, the precoder

* Corresponding author.

E-mail addresses: a.taleshi@nit.ac.ir (A. Alitaleshi), jhamid@nit.ac.ir (H. Jazayeriy), j.kazemitabar@nit.ac.ir (J. Kazemitabar).

matrix indicator, and channel quality indicator are used to determine the target location (Kim et al., 2021). Whereas, RSSI-based positioning systems exploit only collected/received signal strengths (RSS) from several MAC addresses to determine the target location. Therefore, in terms of positioning performance, CSI, by containing more information, shows better stability than the RSSI-based methods (Wang, 2020; Sanam and Godrich, 2020). However, CSI-supported devices require advanced network interface cards, which are usually not embedded in current smartphones (Tiglao et al., 2021). Since RSSI Fingerprinting-based methods do not require additional hardware, they are the most commonly used among other technologies (Yiu et al., 2017).

The Wi-Fi fingerprinting-based positioning methods generally consist of two phases: the offline phase and the online phase (Khajehmehrabi et al., 2017). In the most widely used implementation, each fingerprint sample consists of a vector of received RSS from the observable media access control address (MAC address). In the offline phase, the fingerprint dataset (i.e., radio map) is established by collecting RSS fingerprints at the target area, using site surveying, crowdsourcing, and so on. Each fingerprint has a label, which is the coordinate of the reference point (RP) to which it belongs. This way, the radio map is utilized for model training or pattern matching purposes. In the online phase, the designed positioning model determines the user's location based on the real-time RSS fingerprint.

A key challenge in Wi-Fi fingerprinting-based techniques is how to obtain high-accuracy and low-cost positioning under the lack of learning samples, fluctuation of signal, noise due to device heterogeneity, and multipath effects in complex multi-building/floor environments (Zhu et al., 2020; Raitoharju et al., 2020). Furthermore, the radio map in large-scale environments involving a massive number of MAC addresses/features, feature extraction can efficiently improve positioning accuracy. The exploitation of autoencoders to extract features and reduce dimensions is one of the commonly used techniques in managing a large volume of features. With the rise of deep learning models, deep neural networks (DNN) combined with conventional stacked autoencoders have been widely deployed as a positioning system for large-scale scenarios (Jagannath et al., 2022). The conventional stacked autoencoders (SAEs) are trained with the back-propagation process to optimize the model parameters. The training using back-propagation-based optimization has many problems such as slow convergence, time-consuming, and local minima. Therefore, the training of conventional autoencoders is computationally expensive and time-consuming, especially for large-scale data (Katuwal and Suganthan, 2019a).

A neural network with random weights (NNRW) was proposed to combat the drawbacks of traditional artificial neural networks (Suganthan and Katuwal, 2021). In these networks, weights between the input layer and the hidden layer are adjusted randomly from a given range and kept unchanged during the training process, in contrast, the weights between the hidden layer and the output layer are calculated analytically. NNRW by having a faster training speed compared with traditional learning methods can achieve acceptable accuracy (Zhang and Suganthan, 2016). As types of NNRW, the Random Vector Functional Link Networks (RVFL) is proposed in Pao and Takefuji (1992) in the 1990s. RVFL is a feed-forward network with a single hidden layer and direct links between input and output layers. Simultaneously with the RVFL, Schmidt proposed another single layer feed forward neural network (SLFN) with random weights where, unlike RVFL, there is no direct link between the input layer and output layer. In the Schmidt network, the output weights are determined by using the Fisher method (Schmidt et al., 1992). One of the most attractive theories, Extreme Learning Machine (ELM), was developed in 2004. The ELM can be viewed as a variant of RVFL without the direct links from the input to the output layer and bias term in the output layer, unlike RVFL and Schmidt, respectively (Cao et al., 2017).

However, insufficient training samples can reduce the positioning efficiency of DNN-based methods. On the other hand, although increasing the number of fully connected layers can increase positioning accuracy, it leads to increased computational complexity.

To address the issues mentioned above, this paper proposes a novel deep learning-based framework consisting of the NNRW-based autoencoder and convolutional neural network (CNN) for positioning with Wi-Fi fingerprinting. Compared with the existing indoor positioning methods, the main contributions of this paper are summarized as follows:

1- To localize targets in complex multi-building/floor environments, this paper proposes an innovative deep-learning approach. Two classification and regression models are proposed for floor and 2D-positioning, respectively.

2- The proposed EA-CNN model leverages an extreme learning machine autoencoder by reducing the dimension of the input data. The combination of ELM-AE and CNN, bypassing the computational complexity of conventional SAEs, significantly increases the positioning accuracy.

3- Dealing with training data shortages, this paper proposes a data augmentation strategy by importing noise to the model for increasing the robustness of positioning performance.

4- EA-CNN is evaluated on two available Tampere and UJIIndoor-Loc datasets (Lohan et al., 2017; Torres-Sospedra et al., 2014). The experimental results show the proposed model outperforms the state-of-the-art methods on both floor-level prediction and 2D coordinates positioning.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. Section 3 presents the architecture of the proposed positioning system, including an overall overview of the system and a description of each part of the system. Section 4 describe how to optimize models through experiments on the validation set for both dataset and compares its performance with other fingerprint-based positioning algorithms. Section 5 concludes this work with a discussion on future work.

2. Related work

This section briefly reviews related works to neural networks with random weights and Wi-Fi fingerprinting-based positioning focusing on deep architectures.

2.1. Deep architectures using neural networks with random weights

In recent years, several deep structures using randomization-based autoencoders have been proposed. A multi-layer ELM using randomization-based autoencoders with L2 regularization was suggested in Kasun et al. (2013). By including several autoencoder layers before a classification layer-based ELM, the proposed model obtained better performance than autoencoder-based deep networks and deep belief networks (DBN) for the MNIST dataset. The hierarchical extreme learning machine (HELM) by employing a multi-layer sparse ELM autoencoder is constructed in Tang et al. (2015). In these models, the last layer weights are obtained via a closed-form solution, and the output of autoencoder-based L1 regularization, which is used to generate more sparse and meaningful hidden features can be obtained using the fast iterative shrinkage-thresholding algorithm (FISTA) (Beck and Teboulle, 2009). The several deep RVFL structures utilizing the RVFL framework and randomization-based autoencoders are proposed in Katuwal and Suganthan (2019b). The authors extended the HELM framework by incorporating direct links for feature reusing from different parts of the network to improve the performance of the deep networks-based classification rather than the ELM and shallow RVFL network. An ensemble RVFL network by combining ensemble learning with deep learning was proposed in Shi et al. (2021). The authors in Nayak et al. (2020) propose a deep architect of a stacked random vector functional link-based autoencoder (SRVFL-AE) for multiclass brain abnormalities detection where direct links are used only in the classification layer. The direct links are used only in the classification layer no RVFL-based autoencoder. A convolutional random vector functional link (CRVFL)

neural network is proposed in [Zhang and Suganthan \(2017\)](#), to solve visual tracking problems. In the training process, only fully connected parameters of CRVFL need to be learned, and convolutional filters are randomly initialized and kept unchanged. A hybrid deep architecture using ELM and CNN is proposed in [Zeng et al. \(2015\)](#) to recognize traffic signs. The CNN is trained for feature extraction and ELM is deployed on CNN-learnt features as the classifier. Similar structures are proposed in [Pang and Yang \(2016\)](#), and [Ali et al. \(2020\)](#) for handwritten digit classification.

2.2. Wi-Fi fingerprinting-based positioning using deep learning

Recently, a wide range of machine learning and pattern matching-based methods have been proposed for Wi-Fi fingerprinting-based positioning ([Singh et al., 2021](#)). Neural networks are proposed for fingerprinting-based positioning in [Brunato and Battiti \(2005\)](#). The authors in [Dai et al. \(2016\)](#) proposed a multi-layer feed-forward neural network. Since the mentioned methods have a time-consuming training phase, an extreme learning machine with a single feed-forward layer is utilized to increase the training speed along with improvement of positioning performance ([Lu et al., 2016](#)). Despite their simplicity and easy implementation, they cannot extract useful information from the noisy Wi-Fi signal and deal with signal attenuation, leading to limited accuracy in large-scale buildings. Recently, Deep learning-based neural networks have been successfully deployed in indoor positioning systems, to ameliorate the performance of fingerprinting-based positioning in large-scale environments. They can efficiently explore the critical features by multi-layer feature extraction and get the optimal weights ([Feng et al., 2021; Alhomayani and Mahoor, 2020](#)). In [Nowicki and Wietrzykowski \(2017\)](#) a DNN-based floor and building classification model is proposed to address the labor-intensive and time-consuming issues in Wi-Fi fingerprinting-based positioning. The scalable DNN is proposed in [Kim et al. \(2018\)](#) to estimate building, floor, and floor-level coordinates in the multi-label classification tasks. However, in large-scale environments containing a large number of RPs, the computational complexity of classifiers increases. A CNN-based Wi-Fi fingerprinting method is presented in [Jang and Hong \(2018\)](#) to classify the floor and building by outperforming the DNN-based techniques. To predict the floor, a deep structure of ELM is deployed in [Alitaleshi et al. \(2020\)](#). The authors have utilized a two-layer sparse ELM autoencoder to extract the main features for importing to an ELM-based classification model. The authors in [Alitaleshi et al. \(2022\)](#) have used clustering for grouping overlapping locations. Then, they employed two-label HELM to predict cluster/area and floor in the multi-floor buildings. In [Ezzati Khatab et al. \(2021\)](#), unlabeled data was used to improve the localization performance of a deep-ELM in a single-floor environment.

In [Sinha and Hwang \(2019\)](#) a positioning model using CNN was presented to classify 74 reference points on only a single floor. Their model comprises six layers (four convolutional layers and two fully connected layers with more than 1000 nodes) which can outperform compared CNN application models, such as AlexNet, ResNet, ZFNet, Inception v3, and MobileNet v2. The authors in [Qin et al. \(2021\)](#) have used a positioning system that combines a convolutional denoising autoencoder and a convolutional neural (CDAE) network. The CDAE is used for denoising and extracting critical features from RSSI data. CNNLOC ([Song et al., 2019](#)) has been proposed to hierarchically estimate building, floor, and coordinates by combining SAE and a one-dimensional CNN. In [Oh et al. \(2021\)](#), a newly defined constraint associated with the building boundary is defined to improve the accuracy of the CNNLOC regression model. However, depending on the number of buildings, several models have been trained to predict the target building. Then, the target location is determined by the candidate regression model. DeepLocBox is presented in [Laska and Blankenbach \(2021\)](#) to compute the target area in multi-building/floor environments using Wi-Fi fingerprinting. The authors have used DNN

and CNN to predict a bounding box that contains the user's position. A hierarchical auxiliary deep neural network (HADNN) including multi-tasks learning is proposed in [Cha and Lim \(2022\)](#). The building, floor, and location coordinates are estimated by HADNN optimizing the sum of each loss. The simulation results showed that HADNN achieves better accuracy by setting fewer parameters than the pure two-dimensional regression model. The positioning based on a recurrent neural network (RNN) is proposed in [Ahmed Elesawi and Kim \(2021\)](#) to estimate the target location sequentially from a building to coordinates in terms of classification tasks.

The above methods either do not have significant accuracy in large-scale environments or are computationally expensive. The conventional stacked autoencoders by extraction of meaningful features can increase accuracy in mentioned models. However, the back-propagation process is time-consuming and computationally complex. To simultaneously exploit the advantages of NNRW as autoencoders and CNN capability in hierarchical feature map learning, this work is the first to combine CNN with straightforward extreme learning machine-based autoencoder (ELM-AE) to improve the indoor 3D-positioning accuracy of CNN using Wi-Fi fingerprinting. In addition, this paper has employed an augmentation strategy to deal with data shortage and to improve positioning accuracy in multi-building and multi-floor environments.

3. System overview

This section presents the overall architecture of the proposed positioning system. Then, the description of each portion of the system is provided separately. [Fig. 1](#) shows the proposed system overview. The system is divided into the offline learning phase and the online positioning phase.

In the offline phase, the Wi-Fi fingerprints are collected from training reference points to establish the offline radio map. Then, the data processing stage, which includes data augmentation and data scaling, is performed. Afterward, the clustering algorithm is employed to divide the radio map into two training and validation sets. After training and optimizing, the obtained EA-CNN will be used for real-time positioning. In the online phase, the collected RSSI vector by the user's device after preprocessing is fed into the EA-CNN to predict the target position.

3.1. Data preprocessing

This section describes how to prepare data for the training of the proposed model. The used strategy for data augmentation is presented first, which can take into account noise resulting from environmental changes in the amount of RSS in addition to dealing with the training data shortages. Then, the description of data scaling is described.

3.1.1. Data augmentation

Training a neural network with a small dataset can cause the model to memorize all training instances and lead to an overfitting problem. On the other hand, many environmental factors could influence signal propagation, including the presence of obstacles, moving bodies, signal fluctuation over time, or noise. The existence of a concrete wall causes a weakening of the signal by 5 to 8 dBm. Also, when mobile devices were completely blocked by human bodies, the signal strength would decrease significantly by around 10 dBm ([Xia et al., 2017; Bahl and Padmanabhan, 2000](#)). To take into account the signal fluctuations and to improve the generalization ability of the proposed model, each fingerprint is injected with Gaussian white noise with a zero mean and variance of σ^2 . Then, the noisy data and the original data, are imported into the model as training inputs. To adjust σ^2 , the statistical properties of propagated signals from each MAC in adjacent locations in the test area are analyzed. To estimate the fluctuation of the transmitted signal of each MAC address, first, the variance of each MAC address locally is calculated. The received signal strengths from each MAC at each RP and adjacent locations were used to calculate the local

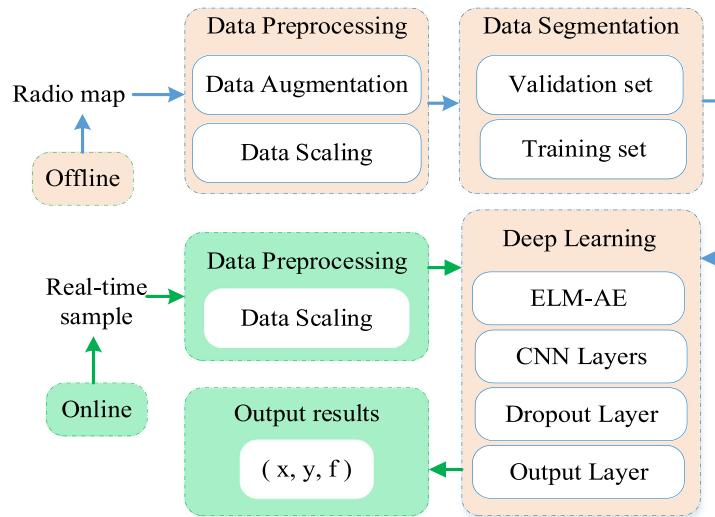


Fig. 1. The overview of the proposed system.

variance of each MAC address. Adjacent locations are those located less than τ meters away from the target RP. Then, by averaging the local variance obtained from all RPs, the global/averaged variance for each MAC address is obtained. This process is described in Algorithm 1. This way, the global variance of each MAC address is estimated, which can be identified with $\mathbf{V}_g = [v_{g,1}, \dots, v_{g,m}] \in \mathbb{R}^{1 \times m}$, where $v_{g,i}$, m have identified the global variance of i th MAC address and the number of detected MAC addresses in the target area. \mathbf{V}_g is used for adjusting the variance of Gaussian noise. In this adjustment, the vector of \mathbf{V}_g consisting variance of each MAC address is to be taken into account for noise generation. Given the radio map, $\Phi \in \mathbb{R}^{N_f \times m}$, consisting of N_f fingerprints obtained from p reference points and m detected MAC addresses, the augmented noisy data, Φ_a , can be obtained as:

$$\Phi_a = \Phi + \text{noise}, \quad \text{noise} = \sqrt{\mathbf{V}_g} \cdot \text{randn}(N_f, m) \in \mathbb{R}^{N_f \times m} \quad (1)$$

Then, the original data is concatenated with augmented data Φ_a to be used in the following steps:

$$\Phi = \text{concatenation}(\Phi_a, \Phi) \quad (2)$$

Algorithm1 The algorithm for calculating the variance of MAC addresses

Inputs: The set of p reference points: $\mathbf{L} = \{l_1, \dots, l_p\}$ $l_i = (x_i, y_i, f_i)$, and the radio map: $\Phi \in \mathbb{R}^{N_f \times m}$

Output: Global variance of MAC addresses: $\mathbf{V}_g \in \mathbb{R}^{1 \times m}$

Steps:

- For each $l_i \in \mathbf{L}$, ($i = 1 : p$):
- Find \mathbf{D} including reference points which are located in the neighborhood less than τ meters away l_i :

$$\mathbf{D} = \{l_j \in \mathbf{L} \mid \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} < \tau, f_i = f_j\}$$

- Calculate the variance of received signals from each MAC in references points are included in \mathbf{D} , and save in $\mathbf{v}_{l_i} = [v_{i,1}, \dots, v_{i,m}] \in \mathbb{R}^{1 \times m}$
- End for
- save all local variances in $\mathbf{V}_l = [\mathbf{v}_{l_1}; \dots; \mathbf{v}_{l_p}] \in \mathbb{R}^{p \times m}$
- Average over the variance of each MAC in p reference points to obtain global variances:

$$\mathbf{V}_g = \{\mathbf{v}_{g,1}, \dots, \mathbf{v}_{g,m}\} \in \mathbb{R}^{1 \times m}, \mathbf{v}_{g,i} = \frac{1}{p} \sum_{j=1}^p \mathbf{v}_{i,j}$$

3.1.2. Data scaling

The value of input raw RSS in both utilized datasets ranges from $-104/-102$ dBm to 0 dBm. The RSS value of any MAC address that is not detected in each fingerprint recording is marked as 100 dBm. Since data normalization is efficient for increasing the learning ability of the neural network, in this paper, data scaling is deployed somewhat similar to Torres-Sospedra et al. (2015), Song et al. (2019), with the difference that each fingerprint is normalized to its maximum instead of the maximum of the whole test and training samples. Furthermore, the authors in Torres-Sospedra et al. (2015) experimentally are shown the powered data tend to represent the RSS values with great accuracy. Accordingly, the simplified power transformation is applied after normalizing each fingerprint by adjusting powering factor to 2. Given radio map $\Phi \in \mathbb{R}^{N_f \times m}$ with N_f fingerprints and m MAC addresses, the i th raw fingerprint can be defined as $\phi_{i,r} = [rss_{i,r}^1, \dots, rss_{i,r}^m]$. Where, $rss_{i,r}^j$ is received signal strength from j th MAC addresses. First, values of 100 are converted to zero, and the normalization process for non- 100 values can be summarized as follows:

- 1- Converting to the positive range:

$$rss_{i,pos}^j = rss_{i,r}^j - (Min - 1), \quad Min = \min_{i,j}(rss_{i,r}^j), \quad j = 1 : m, i = 1 : N_f \quad (3)$$

- 2- Normalization of each fingerprint to its maximum:

$$rss_{j,norm}^i = \frac{rss_{j,pos}^i}{\max_j(rss_{j,pos}^i)}, \quad i = 1 : N_f \quad (4)$$

- 3- Powered transformation:

$$rss_j^i = (rss_{j,norm}^i)^2, \forall i, j \quad \phi_{i,r} \rightarrow \phi_i = [rss_1^i, \dots, rss_m^i] \in \mathbb{R}^{1 \times m} \quad (5)$$

Finally, the scaled radio map $\bar{\Phi} = [\phi_1^T, \dots, \phi_{N_f}^T]^T \in \mathbb{R}^{N_f \times m}$ is utilized for training the neural network.

3.2. Extracting validation set from the training set

In the training phase of the proposed model, several hyperparameters are to be adjusted to control the behavior of the learning process. If these hyperparameters are adjusted based on the training set, overfitting can occur. To avoid overfitting, the validation set is needed to update the model and hyperparameters following the performance of the trained model on the validation set. The random extraction of the validation set is a commonly used method. However, this approach can lead to missing some areas in the indoor positioning problems (Song et al., 2019; Qin et al., 2021). In this paper, a clustering algorithm is utilized to extract the validation set for each floor. First, the training

RPs of each floor is divided into K clusters using the K-Means algorithm. Then, for each cluster, the nearest RP to the cluster head is taken as the validation RP. This process is applied to all floors, and finally, all measured fingerprints of the validation RPs are transferred to the validation set. The remaining samples are considered as the training set. In this study, K is set to be equal to 15% of training reference points on each floor.

3.3. Location estimation model

This section introduces the structure of the proposed model, including feature mapping and positioning model architecture.

3.3.1. Input feature mapping and dimensional reduction using ELM-AE

ELM is a simple training algorithm with a single-layer feed-forward neural network (SLFN) whose parameters are determined fast and efficiently (Bin Huang et al., 2006). The main goal of ELM-AE is to learn the new meaningful data representation in the unsupervised learning task.

Assume N samples with d -dimension are shown as $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_N^T] \in \mathbb{R}^{N \times d}$, where \mathbf{x}_i is the i th sample. First, input data are mapped to a new random space in a predetermined dimension. For the compressed architecture of an ELM-AE with L hidden neurons where $d > L$, the input mapping from d -dimensional space to a L dimensional one can be formulated as (Bin Huang et al., 2006): $\mathbf{h}(\mathbf{x}_i) = g(\mathbf{x}_i \mathbf{w}^T + \mathbf{b}) \in \mathbb{R}^L$, $\mathbf{w}^T \mathbf{w} = \mathbf{I}$, $\mathbf{b}^T \mathbf{b} = 1$. Where $g(\cdot)$ is an activation function, and $\mathbf{w} \in \mathbb{R}^{d \times L}$ is the input weights between the input layer and the hidden layer that are set randomly. The random biases of hidden neurons are identified as \mathbf{b} , and $\mathbf{I} \in \mathbb{R}^L$ is an identity matrix. Then, the outputs of the single-layer feed-forward are computed as $f(\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_i)^T \beta$, $i = 1, \dots, N$. Where $\beta \in \mathbb{R}^{L \times d}$ is the unknown weights matrix, connecting the hidden neurons to the output layer containing N neurons. ELM-AE aims to minimize the reconstruction error of input data. In this way, the loss function of ELM-AE can be formulated as follows (Kasun et al., 2013):

$$\begin{aligned} L_{elm-ae} &= \underset{\beta}{\operatorname{argmin}} \frac{\lambda}{2} \|\mathbf{H}\beta - \mathbf{X}\|_2^2 + \frac{1}{2} \|\beta\|_2^2, \\ \mathbf{H} &= [\mathbf{h}(\mathbf{x}_1)^T, \dots, \mathbf{h}(\mathbf{x}_N)^T]^T \in \mathbb{R}^{N \times L} \end{aligned} \quad (6)$$

Where λ is adjusted for better generalization performance, and the second term controls the complexity of the model. The output weights can be calculated by setting the gradient of L_{elm-ae} with respect to β equal to zero. When the number of training samples N is larger than the number of hidden neurons L , the solution can be obtained in closed-form as follows:

$$\hat{\beta} = (\mathbf{I}/\lambda + \mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{X} \quad N \geq L \quad (7)$$

Where $\mathbf{I} \in \mathbb{R}^N$ is an identity matrix. The learning process of ELM-AE can be summarized in the following three steps:

1. Random allocation of weights and bias \mathbf{w}, \mathbf{b}
2. Calculation of the hidden layer output matrix \mathbf{H}
3. Estimation of the output weights $\hat{\beta}$

When output weights are estimated, mapped input data to new space can be obtained as:

$$\mathbf{X}_{new} = g(\mathbf{X} \times \hat{\beta}^T) \in \mathbb{R}^{N \times L} \quad (8)$$

3.3.2. Positioning with CNN algorithm

The newly mapped data obtained from ELM-AE after converting to 2D virtual images with corresponding labels are imported as input to the CNN framework. In the EA-CNN model, the floor level and the two-dimensional coordinates are predicted separately. The floor level is determined using CNN in terms of the classification task, and a CNN regression model is designed to learn the two-dimensional location.

The convolutional neural networks utilized in this model uses 2D convolutional kernels. After each convolution layer, the “Batch-normalization” layer and “ReLU” activation layer are applied, and the “Dropout” layer is also added in front of the output fully connected layer. A dropout layer randomly sets input elements to zero with a given probability to control overfitting. The local features extracted from the convolution layers (known as feature maps) after passing from the Dropout layer are put into the fully connected layer with N_{oc} , N_{or} neurons in classification and regression models, respectively. In this paper, N_{oc} is equal to the numbers of individual floor levels, and for the position regression model N_{or} is adjusted to be equal to the dimension of 2D location coordinates (x, y). After a fully-connected layer, a widely used Softmax layer is used to convert the output of the fully connected layer into a vector of probabilities. Finally, the classification layer determines the class with the highest probability as the label for the corresponding input. Mean square error and cross-entropy error are used as the loss function for regression and classification layers, respectively. In the real-time positioning phase, the floor level is determined from the trained classification model, and 2D location features (x, y), are obtained from the output of the trained regression model. The overall structure of EA-CNN for classification and regression is illustrated in Fig. 2.

Furthermore, to monitor the training performance of AE-CNN on the validation set, the “Early stopping” strategy is utilized by adjusting the validation patient parameter, ρ . The validation patience is the number of times that validation loss on the trained model does not show improvement compared with the smallest loss so far. This way, the training of the model can be terminated after ρ number of times, and the trained model in the last iteration will be the final model. The hyperparameters containing the number of convolutional layers, kernel size, dropout factor, optimization process, and so on for each dataset will be introduced in the experimental section.

4. Evaluation

This section describes the dataset, the model generation process, and model performance in detail. As performance metrics, classification accuracy and positioning error are utilized. For classification accuracy, the hit rate is calculated for the floor, which counts the correct identification floor labels. The Mean error and the root mean squared error (RMSE) which respectively are the average 2D Euclidean distance and the square root of the average squared distance between predicted and real locations, are used as the positioning error functions.

4.1. Dataset description

Two different testbeds are used for evaluating the performance of our proposed model. Tampere data set is collected in the crowd-sourced strategy (Lohan et al., 2017). In the process of data collection from 4651 locations, 21 mobile devices are used. These providers split samples randomly and in a non-overlapping manner into 697 training fingerprints and 3951 test fingerprints. The number of test samples by having 85% samples compared to 15% training samples, with a huge number of MAC addresses, makes it more challenging than other datasets. Moreover, only one sample is collected in each RP which increases the probability of measurement error. The second testbed is the UJIIndoorLoc dataset which covers three buildings at the University Jaume I, two four-floor buildings, and one five-floor building (Torres-Sospedra et al., 2014). The UJIIndoorLoc contains 21,049 Wi-Fi fingerprint samples collected from 933 RPs with different devices and users. Each fingerprint is identifiable by the location tag, including building number (0, 1, 2), floor level (0–4), and 2D coordinates (x, y). UJIIndoorLoc is divided into two training and test sets, which are collected four months apart. The main features and details of these two datasets are compared in Table 1. Since, in random selection, there is a possibility that there will be more samples from one region than from

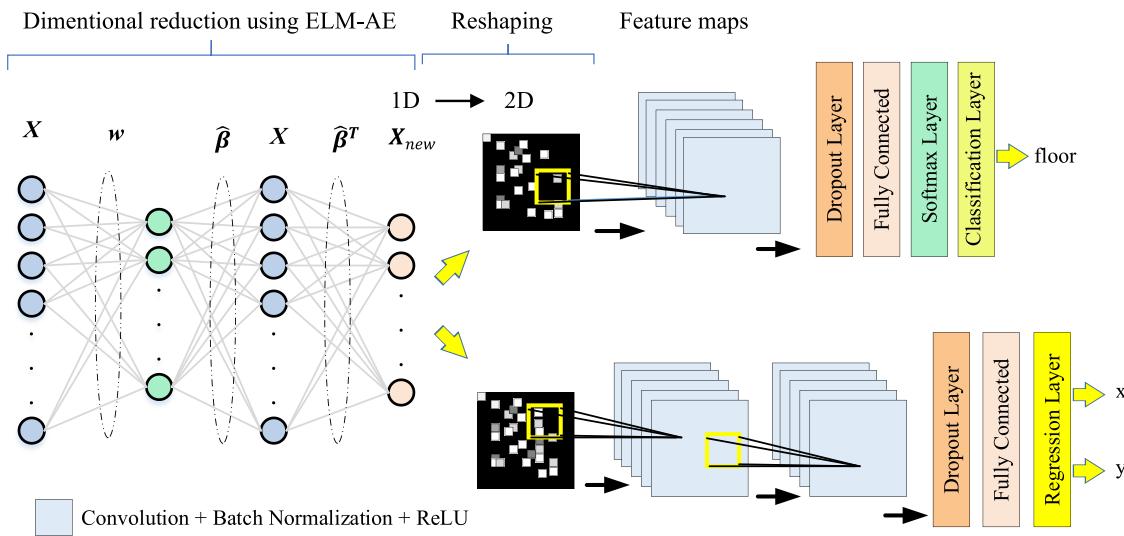


Fig. 2. EA-CNN structure for classification and regression.

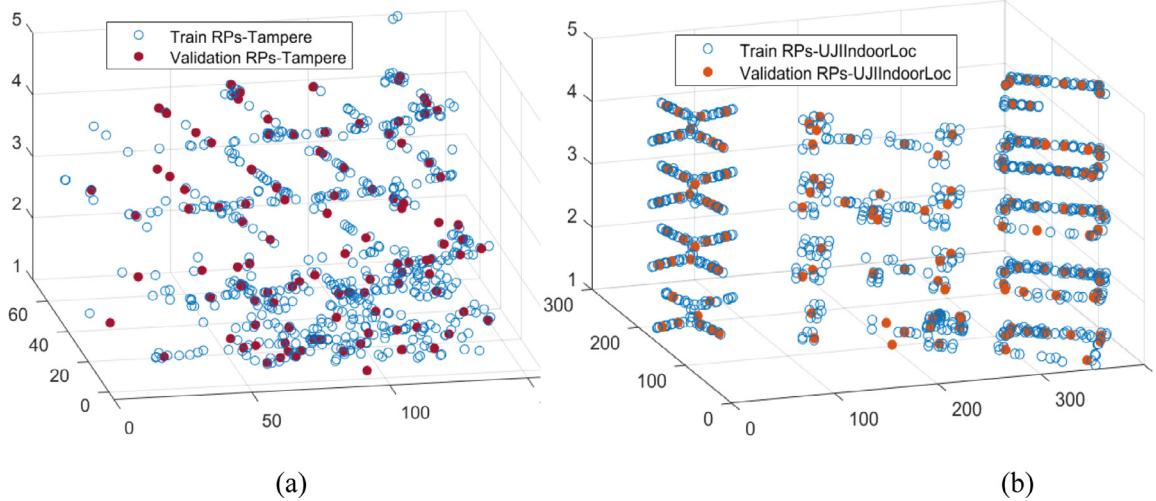


Fig. 3. Training and validation reference points (a) Tampere (b) UJIIndoorLoc.

Table 1

The properties of the datasets used in the evaluation.

Features	UJIIndoorLoc			Tampere
Building ID	B0	B1	B2	B3
Number of floors	4	4	5	5
Number of training RPs	259	265	409	697
Number of training samples	5249	5196	9492	697
Number of test RPs	536	307	268	3951
Number of MAC addresses in training dataset/all	200/520	207/520	203/520	779/992

other regions, K-means clustering is used to cover the entire building from the selected samples. For each data set, clustering is implemented only once at the beginning, and the validation reference points were considered fixed for subsequent experiments. Finally, 15% of training RPs of each floor are selected as a validation set employing mentioned mechanism in Section 3. The obtained training and validation RPs in both datasets are shown in Fig. 3.

In the following, some statistical characteristics of three sets (Train-Validation-Test) in two datasets are evaluated. Fig. 4 shows the histogram of the raw RSS values (in dBm) propagated by MAC addresses in the three sets for two datasets. As shown in Table 2, The standard deviation of the three partitioned sets in the UJIIndoorLoc dataset is

Table 2

Statistical properties of collected raw RSS form all MAC addresses in Tampere and UJIIndoorLoc.

Data set	Mean	Variance	Stdev.
Tampere	Train	-76.96	113.85
	Validation	-77.21	112.09
	Test	-76.97	111.79
UJIIndoorLoc	Train	-78.65	157.69
	Validation	-77.77	165.22
	Test	-77.45	145.22

around 10 and in the Tampere dataset, it is about 12. Although the average RSS in the two datasets is in the same range, the Tamper dataset has more uncertainty, and its variance is more than the UJIIndoorLoc dataset due to recording only one sample in each reference point.

4.2. Experiments evaluation

This paper experimentally explores how to generate and then optimize the AE-CNN model. The parameters used in the optimization process and fixed parameters are shown in Table 3. The activation functions in ELM-AE and CNN models are the sigmoid function and

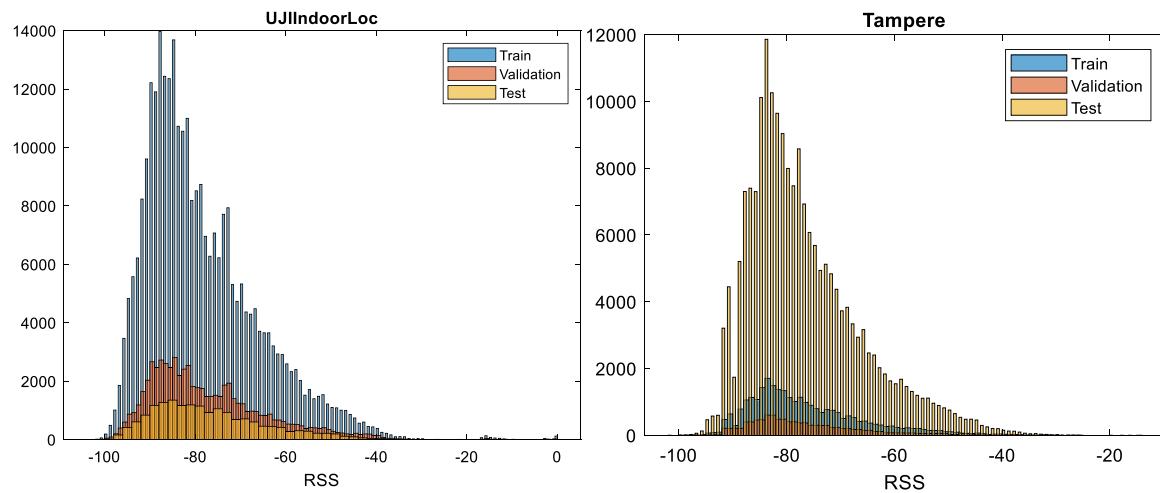


Fig. 4. Histogram of the raw RSS values (in dBm).

Table 3
Model parameters.

	Parameters	Value
Fixed	Max epoch	50
	Mini batch size	90
	Padding method	Zero and Same
	Shuffle	Every-epoch
	The activation function of CNN	ReLU
	The activation function of ELM-AE	Sigmoid
	Initial learning rate	0.01
	Learning rate drop factor	0.1
	Learning rate drop period	10
	Autoencoder parameter, λ	0.01
Adjusted in the optimization process	Optimizer	ADAM, SGDM
	Autoencoder size	70–272
	Number of convolution layers	1, 2
	Filter size	3, 5, 7
	Number of filters	32, 64, 96, 128
	Dropout factor	0.2–0.7
	L2 regularization factor	0.1, 0.01, 0.001

the Rectified Linear Unit (ReLU), respectively. The patience parameter in early stopping, ρ , is set to 5. It is worth mentioning, that the reported results in the training phase are obtained by averaging over the results of several repetitions. All the experiments are implemented with Matlab-2021b deep learning toolbox, and the simulations are run on a computer with Core i7-7500 CPU, 8G RAM, NVIDIA GeForce 940MX GPU.

4.2.1. EA-CNN model optimization for floor estimation

Since in the Tampere dataset, floor height in meters is used instead of floor number, a preprocessing is required to present the floor levels as categorical labels (1, 2, ..., 5) for importing the classification model. In the UJIIndoorLoc dataset, each sample is labeled to (x, y, floor level, building ID). To classify floors using the EA-CNN, paired floor level and buildings ID are merged to make a new label to identify 13 separate floors in the whole target area. For example, a collected sample in building 1 on floor 3 is categorized as the seventh class. Since in the UJIIndoorLoc dataset, unlike the Tampere dataset, an average of twenty samples were collected at each RP, to evaluate the efficiency of proposed data augmentation, only some of the training samples, including 25% of samples in each RP are considered for model training and ignored the rest.

A. Convolution layer and ELM-AE parameters adjustment

The proposed classification model consists of one convolution layer, and the remaining parameters in a hierarchical approach are optimized. First, to determine filter parameters (size, number), the autoencoder is

Table 4

Validation performance in terms of various filter parameters for floor classification for two Tampere and UJIIndoorLoc dataset.

Convolution layer	Validation Floor hit rate (%)	
	Tampere	UJIIndoorLoc
Filter Size – Number		
3,3 – 64	88.99	98.65
3,3 – 96	88.07	97.98
3,3 – 128	88.99	96.64
5,5 – 64	90.82	97.85
5,5 – 96	90.82	98.25
5,5 – 128	92.66	98.94
7,7 – 64	90.82	97.71
7,7 – 96	92.66	97.18
7,7 – 128	90.82	97.31

initialized to a fixed value, the Dropout factor is initialized to 0.2, and the “ADAM” method is set for optimization. To adjust the parameters of the convolution layer, the autoencoder hidden neurons are initialized to the number of MAC addresses whose frequency of hearing was higher than average. Therefore, in Tampere and UJIIndoorLoc, the autoencoder size is set to 272 and 156, respectively. This way, each sample, after dimensional reduction using ELM-AE and normalization, is transformed into 16×17 and 12×13 virtual images to be within [0,1]. Then, the performance of the obtained model is evaluated on the validation set (without augmentation), changing filter size from 3 to 7 for different filter numbers from 64 to 128. It is worth mentioning, that the reported results for each parameter’s tuning have been obtained by averaging over the results of several repetitions. The two datasets are different in terms of class and feature numbers. The UJIIndoorLoc, with 13 distinguished floors, has more classes than the TU with 5 floors, while the number of UJIIndoorLoc features is less than TU. On the other hand, TU has fewer training samples. Therefore, it is expected that these cases will affect the determination of parameters and different results will be obtained in terms of positioning accuracy. As shown in Table 4, regardless of the number of filters, when the filter size is 5, the classification accuracy is higher in both datasets. Increasing the filter size, despite the increase in learning parameters does not lead to a significant improvement. Given the filter size of 5, when the number of filters is set to 128, relatively better accuracy is obtained. Therefore, these values are set for the following optimization steps.

Afterward, model performance is evaluated by decreasing the number of ELM-AE hidden neurons. The evaluation result is depicted in Fig. 5. In the Tampere dataset, when the number of ELM-AE hidden neurons is set to 81, the accuracy of floor identification reaches 92, 95%, and in UJIIndoorLoc data, when the number of autoencoder neurons is set to 64, the accuracy of the validation data has the

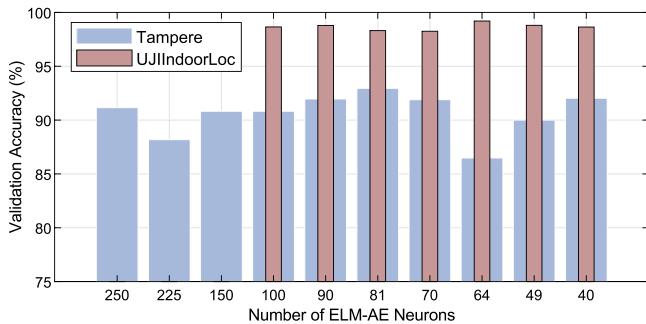


Fig. 5. Validation performance in terms of different ELM-AE hidden neurons for floor classification.

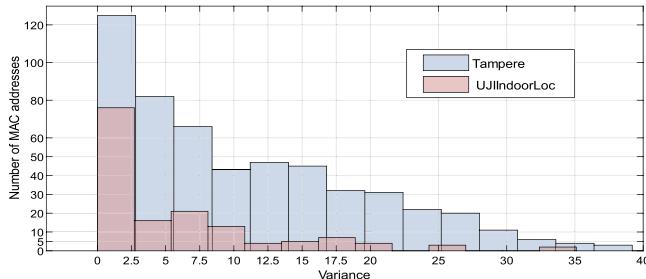


Fig. 6. Histogram of global variance of collected RSS in detected MAC addresses in the training samples.

highest value. As the results show, since the number of MAC addresses in Tamper is higher than UJIIndoorLoc, the need for more neurons for more meaningful mapping to the new space can be justified. The accuracy of the UJIIndoorLoc dataset related to changing the number of neurons is more stable than the TU. Nevertheless, obtaining the highest accuracy is considered the final selection criterion.

B. Data augmentation, applying Dropout, and optimizer setting

As mentioned in Section 3.1, a mechanism for data augmentation is presented. Here, the efficiency of the proposed method on the performance of the obtained model is evaluated. The augmentation process is run in several rounds according to Algorithm 1, and for each round, one new radio map is generated. The value of τ is adjusted to be 1.5 meters for analyzing the local variance of the signals propagated from MAC addresses. Fig. 6 shows the histogram of the global variance of collected RSS from MAC addresses that are heard more than others in two datasets. According to the results, the average variance in the Tampere dataset is 7.79 dBm, which is 65.46% higher than the obtained average variance in the UJIIndoorLoc training dataset (2.69 dBm). The higher variance in the TU dataset may be due to the recording of only one sample at each reference point, which increases the possibility of uncertainty and measurement error.

The obtained augmented radio map will be concatenated with the original data to be fed to the model for training in the following experiments. As shown in Table 5, in both datasets, by data augmentation after two rounds, validation accuracy can reach its highest value. Due to the smaller number of training samples in the TU dataset, accuracy improvement is more noticeable with the application of data aggregation. However, in the UJIIndoorLoc, the validation accuracy has not improved significantly after increasing the data. In other words, including more samples is not very effective in increasing the accuracy of floor classification, and having three to five samples is enough.

To avoid overfitting in the deep learning model and further generalize the proposed model, a two-parameter dropout factor and L2-regularization factor are utilized. L2 regularization penalizes the sum of the squared weights to control the model complexity by penalizing

Table 5

Comparison of augmentation efficiency on classification performance of validation data for different rounds (iterations).

Rounds number of Augmentation	Floor hit rate (%)	
	Tampere	UJIIndoorLoc
1	93.64	98.75
2	95.50	99.29
3	90.35	98.05

higher terms in the model. The dropout layer randomly drops connections in training sessions with predefined probability (or factor). The model performance is evaluated by varying the Dropout factor and L2-regularization factor in a training stage with two optimizers, including adaptive moment estimation (ADAM), and stochastic gradient descent with momentum (SGDM). The experimental results on the Tampere dataset are presented in Table 6. Regardless of the optimization method and the L2-regularization factor, the classification accuracy is highest when the drop-out is 0.4. By applying L2-regularization and dropout, more accuracy is obtained when the L2 value is one of the two values 0.01 or 0.001. What can be deduced from the results is that SGDM has better performance than ADAM in most cases. When the Dropout factor is set to 0.4, the SGDM reaches the highest accuracy if L2-regularization equals 0.001. On the other, the same accuracy can be achieved by ADAM when the Dropout factor and L2-regularization factor are set to 0.5 and 0.001, respectively. As shown, using the patience parameter in the early stopping strategy causes the training loop to stop before reaching the maximum epoch. In the following, these experiments are performed on the UJIIndoorLoc dataset and obtained mostly similar results. The best result is obtained when the optimizer is SGDM and Dropout is set to 0.4.

4.2.2. EA-CNN model optimization for 2D positioning

Using the obtained classification model, EA-CNN can determine the position of users in terms of the floor level. To determine the absolute coordinates, an EA-CNN-based regression model should be generated.

A. Convolution layer and ELM-AE parameters adjustment

The structure and initial parameters of the EA-CNN regression model are mainly the same as the floor classification model. Since in our experiments, using one convolution layer to estimate absolute coordinates does not achieve significant initial accuracy, the EA-CNN regression model is initialized to two convolutional layers. According to the experience gained from the floor classification settings, to adjust the number of ELM-AE hidden neurons and filter properties, the number of ELM-AE hidden neurons is selected to be {200, 100, 90, 81}, and {90, 81, 70, 64} for Tampere and UJIIndoorLoc testbeds, respectively. For each case, model performance is evaluated by changing filter size.

The results are summarized in Table 7. In both datasets, and regardless of the number of autoencoder neurons, the positioning error mostly decreases with the increasing size and number of filters. Similar to the classification model, implementation of ELM-AE with 81 and 64 hidden neurons leads to better performance in Tampere and UJIIndoorLoc datasets, respectively. Among examined filter sizes in this situation, {(7,7-96), (7,7-64)} for Tampere and {(7,7-96), (7,7-96)} for UJIIndoorLoc achieve the better performance. The number of more filters in the UJIIndoorLoc dataset can be related to its larger scale of the environments than the TU dataset.

B. Data augmentation, Dropout, and optimizer setting

To evaluate the efficiency of the augmentation method on the performance of the obtained regression model, experiments same as 4.2.1 section are examined. Table 8 shows obtained experimental results. When augmentation is run in two rounds by considering V_g , the Mean error is reduced by 1.2 meters in the Tampere testbed. The experiments on the UJIIndoorLoc validation set demonstrate that the proposed model will perform better when the number of augmentation rounds is three. In contrast to the classification model, the positioning accuracy

Table 6

Effects of Dropout, L2-regularization factor, and optimizers on the performance of EA-CNN for classification of floors in Tampere validation data.

Parameter	Optimizer	ADAM		SGDM	
		Floor hit rate (%)	Stop Epoch	Floor hit rate (%)	Stop Epoch
0.2	–	95.49	24	96.09	32
		92.79	47	94.59	45
		95.79	38	96.09	32
		94.59	41	96.39	38
0.3	–	94.89	67	95.79	35
		89.49	18	94.89	18
		96.09	35	96.09	38
		94.89	55	94.89	35
0.4	–	95.49	24	96.39	32
		93.39	36	96.59	18
		95.49	35	96.09	44
		94.89	38	96.69	47
0.5	–	94.29	27	96.50	29
		92.79	35	94.90	18
		95.49	29	96.39	44
		96.69	44	96.39	32
0.6	–	95.19	32	95.49	21
		93.09	41	94.60	18
		95.49	32	96.40	55
		95.20	29	95.80	38
0.7	–	95.19	27	95.49	67
		92.79	35	93.99	18
		96.10	29	95.49	38
		96.39	41	96.09	47

Table 7

Validation performance in terms of various filter parameters along with a various number of ELM-AE hidden neurons for 2D positioning.

ELM-AE neurons	Convolution layer	Positioning error (m)			
		Tampere		UJIIndoorLoc	
		Mean	RMSE	Mean	RMSE
Tampere/ UJI	Filter Size – Number	1th-layer	2th-layer		
100/ 90	3,3 – 96	3,3 – 64	11.51	18.16	10.30
	5,5 – 96	5,5 – 64	11.24	17.82	9.40
	7,7 – 96	7,7 – 64	11.28	18.16	9.10
	7,7 – 96	7,7 – 96	11.15	17.41	8.9
90/ 81	3,3 – 96	3,3 – 64	11.81	18.73	10.55
	5,5 – 96	5,5 – 64	11.5	18.58	9.31
	7,7 – 96	7,7 – 64	11.33	17.92	8.9
	7,7 – 96	7,7 – 96	11.54	18.65	8.75
81/ 64	3,3 – 96	3,3 – 64	11.06	15.13	9.93
	5,5 – 96	5,5 – 64	10.99	15.08	9.14
	7,7 – 96	7,7 – 64	10.61	14.69	8.84
	7,7 – 96	7,7 – 96	10.70	14.7	8.73
					12.71

Table 8

Comparison of data augmentation efficiency on 2D-positioning performance on validation data in terms of different rounds number.

Rounds number of Augmentation	Positioning Error (m)			
	Tampere		UJIIndoorLoc	
	Mean	RMSE	Mean	RMSE
1	10.41	15.85	7.64	9.57
2	9.48	14	7.54	10.47
3	9.98	15.37	6.70	8.79

by reducing Mean error up to 22.98% and RMSE up to 30.78% is more noticeable after applying data augmentation.

The EA-CNN regression model is also optimized in terms of the Dropout factor and L2-regularization factor. Although changing these parameters did not lead to much improvement in our experiments on both datasets, however, the best accuracy with a Mean error of 9.32, 6.65 meters in Tampere and UJIIndoorLoc datasets can be achieved by ADAM optimizer when Dropout and L2-regularization are set to 0.2

and 0.01, respectively. Therefore, these values are adjusted in the final regression model.

4.2.3. Experiments on test data

In this section, the performance of the proposed model on the test data is evaluated. The obtained optimal parameters of EA-CNN for both classification and regression models are summarized in Table 9. The random and optimal weights obtained from the final ELM-AE along with the weights of the convolution network are stored and used for predicting the location of the test sample. Similar to the training stage, each test sample in real-time after scaling and passing from trained ELM-AE is reshaped to the 2D-normalized image. Then, the trained models predict its floor level and 2D coordinates. The proposed model can predict floor level with an accuracy of 94.53%, and 96.13% and achieves 8.24, and 8.91 meters average positioning error in Tampere and UJIIndoorLoc datasets, respectively.

This section describes how to obtain the three data representation methods for RSSI results in the smallest mean positioning error. The positioning error and floor hit rate obtained by each of the three presentations are compared in Table 10, while the network parameters are

Table 9

The EA-CNN optimal parameters and their impact on Tampere test data performance.

Optimal parameters	Floor hit rate (%) and Positioning Error (m)							
	Name	Tampere/UJIIndoorLoc		Tampere			UJIIndoorLoc	
		Classification	Regression	Floor	Mean	RMSE	Floor	Mean
Autoencoder size		81/64	81/64					
Number of convolution layers		1	2					
Filter size		5,5	7,7					
Number of filters		128	96, 64/96, 96					
L2-regularization factor		0.001	0.01	94.53	8.24	11.06	96.13	8.91
Number of Augmentation rounds		2	2/3					
Optimizer		SGDM	ADAM					
Dropout factor		0.4	0.2					

Table 10

Comparison of the effect of data representation on positioning performance in UJIIndoorLoc and Tampere datasets.

Data set	Accuracy	Data scaling		
		Raw	Min-max normalized	Powered transformed
UJIIndoorLoc	Floor hit rate (%)	78.57	93.81	96.13
	Mean (m)	13.38	9.49	8.91
	RMSE (m)	17.70	12.85	11.73
TU	Floor hit rate (%)	87.64	93.87	94.53
	Mean (m)	11.03	8.40	8.24
	RMSE (m)	13.87	11.37	11.06

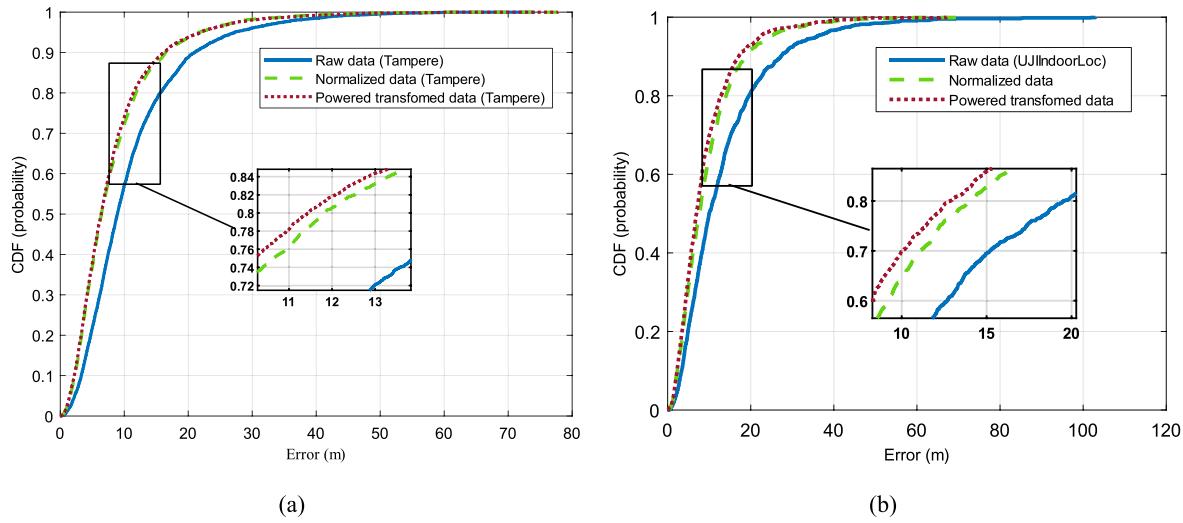


Fig. 7. CDF comparison of positioning error for three data representations. (a) Tampere dataset (b) UJIIndoorLoc dataset.

similar to [Table 9](#). CDF comparison of positioning error of regression model for three data representations is depicted in [Fig. 7](#). According to the results of the experiments, using Powered transformed RSS data can achieve better accuracy in terms of floor hit rate and positioning in both datasets. The positioning accuracy in Tampere dataset is almost the same for normalized and powered transformed samples. Whereas, in the UJIIndoorLoc dataset, the use of powered transformed scaling leads to more improvement in floor classification.

In the following, we were inclined to examine what effect it would have on the test results if we increased the number of rounds of data augmentation. In this way, the effect of data augmentation is evaluated on the EA-CNN performance in the Tampere test dataset, by increasing of round numbers up to 8. In this stage and following, both classifier and regression models have trained again on the new obtained dataset for each augmentation round. All parameters of the two models, except the number of augmentation rounds, are following [Table 9](#). It is noteworthy that, since the hidden neurons of the ELM-AE layer have not changed, the weights of the first layer are the same as the optimal models. The results show that, although in the training phase, adding the number of rounds more than two or three times, does not improve

the result, it leads to an increase in the accuracy in the test phase. For distinguishing, the higher accuracy obtained from these experiments was named the best performance.

As shown in [Table 11](#), by increasing the rounds, the positioning performance of the EA-CNN improves and by performing seven rounds, it reaches maximum accuracy. Furthermore, the positioning performance can be improved up to 16.45% by data augmentation compared to when only original data is used for model training.

As mentioned previously, EA-CNN is trained and optimized with 25% of samples of each RP by applying two rounds of augmentation in UJIIndoorLoc. To investigate how changing the ratio affects the positioning performance, different values were tested. The training samples ratio was changed from 5% to 100% without data augmentation. The model was trained based on different data ratios and the positioning performance of EA-CNN is evaluated on test samples. The results are listed in [Table 12](#). The experiment results demonstrate that the best performance of the proposed AE-CNN can be obtained when 100% of samples are used for the training of the model. Also, when the ratio is 25%, deploying of proposed augmentation can reduce average positioning error up to 11.45% from 9.93 to 8.91-meters.

Table 11

Comparison of data augmentation efficiency on positioning performance of EA-CNN in terms of Tampere test dataset.

Number of augmentation rounds	Floor hit rate (%) and Positioning Error (m)		
	Floor	Mean	RMSE
3	94.83	8.39	11.64
4	94.80	8.31	11.22
5	94.91	8.22	11.36
6	95.09	7.98	10.76
7	95.30	7.96	10.71
8	94.78	8.17	11.30
Without augmentation	94.17	9.27	12.08

Table 12

Comparison of positioning performance of optimized EA-CNN by changing training samples ratio in UJIIndoorLoc dataset without data augmentation (*Ratio of the number of utilized samples to available samples in each RP for model training*).

Ratio	Floor hit rate (%) and Positioning Error (m)		
	Floor	Mean	RMSE
5%	96.03	15.16	19.25
25%	96.13	9.93	13.10
50%	96.22	8.95	11.67
75%	96.31	8.61	11.05
100%	96.31	8.34	10.72

4.3. Comparison with state-of-the-art methods

Table 13 compares the positioning performance on the Tampere and UJIIndoorLoc dataset between the EA-CNN-based system and some deep learning-based positioning methods. The performance of the improved model is reported in addition to the best results from which EA-CNN is obtained. The optimal model is structured in the training phase, and the best performance is related to the best result obtained in the testing phase after changing only the data size. All the compared

methods have been trained and evaluated on the same training and test data, whereas, the optimal model in UJIIndoorLoc is obtained from only 25% of training samples. To make a more detailed comparison with the proposed method, first, a summary of the structure of the mentioned systems is described.

EA-CNN in addition to comparing with the basic CNN model without ELM-AE has been compared with four existing CNN-based methods, CDAE-CNN (Qin et al., 2021), CNNLOC (Song et al., 2019), C-CNNLOC (Oh et al., 2021), and DeepLocBox (Laska and Blankenbach, 2021) and some other deep learning models, including direct-ELM/HELM (Alitaleshi et al., 2022), DNN (Scalable DNN (Kim et al., 2018), HADNN (Cha and Lim, 2022)) and RNN (Ahmed Elesawi and Kim, 2021). The CDAE-CNN and CNNLOC consist of two classification models and one regression model for building, floor, and coordinate estimation. The C-CNNLOC consists of a building classification model and contains several regression models, which are trained on each building, separately. The DeepLocBox has only one regression model that includes floor and building information in addition to two-dimensional coordinates. The direct-ELM is similar to original ELM classifiers, which included 400 and 300 neurons for Tampere and UJIIndoorLoc datasets, respectively. The direct-HELM uses a hierarchical structure, including two ELM sparse-autoencoder layers and one classification layer with (200–90–400) and (90–90–300) neurons for Tampere and UJIIndoorLoc datasets respectively (Alitaleshi et al., 2022). The two models Scalable DNN, and HADNN, which consist of an autoencoder and fully connected layers, are modeled in the form of multi-label classification and regression tasks.

Unlike structures such as (Sinha and Hwang, 2019), which do not have autoencoders, our model aims to increase the speed of convolution operation by creating smaller images, which leads to a computational cost reduction. On the other hand, the models presented in CNNLOC, C-CNNLOC, CDAE-CNN, and HADNN due to the use of two or three stacked autoencoder layers, have more learning parameters and computational complexity compared to the EA-CNN including only one

Table 13

Comparison of EA-CNN positioning performance with state-of-the-art methods (no-reported results in benchmark algorithms are marked with a dash “-”).

Dataset	Method	Floor hit rate (%) and Positioning Error (m)		
		Floor	Mean	RMSE
Tampere	CNNLOC (Song et al., 2019)	94.22	10.88	-
	HADNN (Cha and Lim, 2022)	94.58	9.05	-
	direct-ELM (Alitaleshi et al., 2022)	89.27	10.91	15.62
	direct-HELM (Alitaleshi et al., 2022)	91.21	9.10	13.99
	CNN (without ELM-AE)	94.10	11.22	14.54
	EA-CNN	Improved model	94.53	8.24
		Best performance	95.30	7.96
UJIIndoorLoc	CNNLOC (Song et al., 2019)	96.03	11.78	-
	C-CNNLOC (Oh et al., 2021)	-	8.87	-
	CDAE-CNN (Qin et al., 2021)	95.30	12.4	-
	DeepLocBox (Laska and Blankenbach, 2021)	92.62	9.07	-
	Scalable DNN (Kim et al., 2018)	91.27	9.29	-
	HADNN (Cha and Lim, 2022)	93.15	14.93	-
	RNN (Ahmed Elesawi and Kim, 2021)	Standard	94.42	8.68
		LSTM	95.23	8.66
	direct-ELM (Alitaleshi et al., 2022)	90.45	10.57	14.77
	direct-HELM (Alitaleshi et al., 2022)	93.60	8.42	12.5
	CNN (without ELM-AE)	95.14	11.99	15.57
	EA-CNN	Improved model	96.13	8.91
		Best performance	96.31	8.34

autoencoder layer. Moreover, due to the lack of the need for a back-propagation process and iteration-based optimization, ELM-AE has less training time than stacked autoencoders utilized in compared models. In this way, the proposed model has a lighter structure and a faster training process. Furthermore, the proposed system has only used two models to estimate the location in environments containing multiple buildings. Indeed, without the need for a separate model to determine the building (such as CNNLOC, C-CNNLOC, CDAE-CNN) it can be interpreted from the output floor number.

First, by removing the ELM-AE part, the positioning performance of only the CNN model is evaluated. Using ELM-AE in addition to reducing input dimension and extracting meaningful features, average positioning error can be decreased by up to 40.95%, and RMSE by up to 35.76% in the Tampere dataset, and the UJIIndoorLoc dataset, it reached up to 43.74% and 45.24%. Compared to the hierarchical structure in the direct-HELM, the shallow ELM has lower accuracy, and the proposed EA-CNN model overcomes both methods in terms of floor detection and positioning accuracy. In the Tampere dataset, EA-CNN having a lighter structure than CNNLOC (including two layers stacked autoencoder, three convolutional layers, and one fully connected layer for both floor and coordinates estimation), has obtained lower positioning error in both cases with and without data augmentation. Here, the improved model and best performance have been obtained after two and seven rounds of augmentation, respectively. It can be found that the EA-CNN system reduces the average positioning error by up to 36.68%, compared to other existing methods. On the UJIIndoorLoc dataset, the optimized EA-CNN model improves the positioning performance by 67.56% compared to the mentioned methods by exploiting only 25% of the training samples. Meanwhile, the introduced methods have used all the training samples. In terms of floor estimation, EA-CNN outperforms all the compared methods.

5. Conclusions

This paper proposes a Wi-Fi fingerprint positioning system using ELM autoencoder and CNN. The ELM-AE can reduce and extract features from RSSI data, which effectively improves the positioning accuracy of CNN. To deal with data shortages and improve model robustness and positioning performance, this paper has utilized an augmentation strategy. The performance of EA-CNN has been evaluated on two Tampere and UJIIndoorLoc datasets. The experiment results demonstrate that EA-CNN classification-based model predicts floor with higher accuracy than the compared method by 96.31% and 95.30% on UJIIndoorLoc and Tampere, respectively. Moreover, EA-CNN regression-based model can achieve better performance in terms of average positioning error (by 8.34 and 7.96-meters on UJIIndoorLoc and Tampere) compared to CNN and state-of-the-art methods in multi-building and multi-floor environments. In future work, we intend to evaluate the performance of the proposed model by changing the cost function in the regression model by adding new constraints and taking into account the simultaneous estimation of the floor. This can be deployed by the definition of new constraints to penalize incorrect floor estimates and out-of-dimension estimates of the experimental area. We are also interested in including other types of NNRW in our modeling. In addition, we will study the fusing of Wi-Fi with other signals such as magnetic fields and Bluetooth to create multi-channel images for learning by the proposed model.

CRediT authorship contribution statement

Atefe Alitaleshi: Conceptualization, Methodology, Software, Visualization, Writing – original draft. **Hamid Jazayeriy:** Supervision, Investigation, Validation, Writing – review & editing. **Javad Kazemitabar:** Supervision, Investigation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The supplementary files associated with this paper (code, input files, etc) will be made available on request.

References

- Ahmed Elesawi, A.E., Kim, K.S., 2021. Hierarchical multi-building and multi-floor indoor localization based on recurrent neural networks. In: Proc. - 2021 9th Int. Symp. Comput. Netw. Work. CANDARW 2021. pp. 193–196. <http://dx.doi.org/10.1109/CANDARW53999.2021.00038>.
- Alhomayani, F., Mahoor, M.H., 2020. Deep learning methods for fingerprint-based indoor positioning: A review. *J. Locat. Based Serv.* 14 (3), 129–200. <http://dx.doi.org/10.1080/17489725.2020.1817582>.
- Ali, S., Li, J., Pei, Y., Aslam, M.S., Shaikat, Z., Azeem, M., 2020. An effective and improved CNN-ELM classifier for handwritten digits recognition and classification. *Symmetry* 12 (10), 1742. <http://dx.doi.org/10.3390/SYM12101742>, 12, 1742, 2020.
- Alitaleshi, A., Jazayeriy, H., Kazemitabar, S.J., 2020. WiFi fingerprinting based floor detection with hierarchical extreme learning machine. In: 2020 10h Int. Conf. Comput. Knowl. Eng.. ICCKE 2020, pp. 113–117. <http://dx.doi.org/10.1109/ICCKE50421.2020.9303624>.
- Alitaleshi, A., Jazayeriy, H., Kazemitabar, J., 2022. Affinity propagation clustering-aided two-label hierarchical extreme learning machine for Wi-Fi fingerprinting-based indoor positioning. *J. Ambient Intell. Humaniz. Comput.* 2022 136 13 (6), 3303–3317. <http://dx.doi.org/10.1007/S12652-022-03777-1>.
- Bahl, P., Padmanabhan, V.N., 2000. Radar: An in-building RF-based user location and tracking system. In: Proceedings - IEEE INFOCOM, Vol. 2. pp. 775–784. <http://dx.doi.org/10.1109/infcom.2000.832252>.
- Basiri, A., Lohan, E.S., Moore, T., Winstanley, A., Peltola, P., Hill, C., Amirian, P., Silva, P.F.E., 2017. Indoor location based services challenges, requirements and usability of current solutions. *Comput. Sci. Rev.* 24, 1–12. <http://dx.doi.org/10.1016/J.COSREV.2017.03.002>.
- Beck, A., Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm. *Soc. Ind. Appl. Math. J. Imaging Sci.* 2 (1), 183–202. <http://dx.doi.org/10.1137/080716542>.
- Bin Huang, G., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: Theory and applications. *Neurocomputing* 70 (1–3), 489–501. <http://dx.doi.org/10.1016/j.neucom.2005.12.126>.
- Brunato, M., Battiti, R., 2005. Statistical learning theory for location fingerprinting in wireless LANs. *Comput. Netw.* 47 (6), 825–845. <http://dx.doi.org/10.1016/J.COMNET.2004.09.004>.
- Cao, W., Wang, X., Ming, Z., Gao, J., 2017. A review on neural networks with random weights. pp. 1–10. <http://dx.doi.org/10.1016/j.neucom.2017.08.040>.
- Cha, J., Lim, E., 2022. A hierarchical auxiliary deep neural network architecture for large-scale indoor localization based on Wi-Fi fingerprinting. *Appl. Soft Comput.* 120, 108624. <http://dx.doi.org/10.1016/j.asoc.2022.108624>.
- Dai, H., Ying, W.H., Xu, J., 2016. Multi-layer neural network for received signal strength-based indoor localisation. *IET Commun.* 10 (6), 717–723. <http://dx.doi.org/10.1049/IET-COM.2015.0469>.
- Dumbgen, F., et al., 2019. Multi-modal probabilistic indoor localization on a smartphone. In: 2019 Int. Conf. Indoor Position. Indoor Navig.. IPIN 2019, <http://dx.doi.org/10.1109/IPIN.2019.8911765>.
- Ezzati Khatab, Z., Hajihoseini Gazestani, A., Ghorashi, S.A., Ghavami, M., 2021. A fingerprint technique for indoor localization using autoencoder based semi-supervised deep extreme learning machine. *Signal Process.* 181, 107915. <http://dx.doi.org/10.1016/J.SIGPRO.2020.107915>.
- Feng, X., Nguyen, K.A., Luo, Z., 2021. A survey of deep learning approaches for WiFi-based indoor positioning. <http://dx.doi.org/10.1080/24751839.2021.1975425>.
- Geok, T.K., et al., 2021. Review of indoor positioning: Radio wave technology. *Appl. Sci.* 11 (1), 1–44. <http://dx.doi.org/10.3390/app11010279>.
- Jagannath, A., Jagannath, J., Kumar, P.S.P.V., 2022. A comprehensive survey on radio frequency (RF) fingerprinting: Traditional approaches. *Deep Learn. Open Challenges* 14 (8), 1–30.
- Jang, J.W., Hong, S.N., 2018. Indoor localization with WiFi fingerprinting using convolutional neural network. In: Int. Conf. Ubiquitous Futur. Networks. ICUFN, 2018-July, pp. 753–758. <http://dx.doi.org/10.1109/ICUFN.2018.8436598>.
- Kasun, L.L.C., Zhou, H., Bin Huang, G., Vong, C.M., 2013. Representational learning with ELMs for big data. *IEEE Intell. Syst.* 28 (6), 31–34.
- Katuwal, R., Suganthan, P.N., 2019a. Stacked autoencoder based deep random vector functional link neural network for classification. *Appl. Soft Comput.* 85 (xxxx), 105854. <http://dx.doi.org/10.1016/j.asoc.2019.105854>.

- Katuwal, R., Suganthan, P.N., 2019b. Stacked autoencoder based deep random vector functional link neural network for classification. *Appl. Soft Comput.* 85, 105854. <http://dx.doi.org/10.1016/j.asoc.2019.105854>.
- Khalajmehrabadi, A., Gatsis, N., Akopian, D., 2017. Modern WLAN fingerprinting indoor positioning methods and deployment challenges. *IEEE Commun. Surv. Tutor.* 19 (3), 1974–2002. <http://dx.doi.org/10.1109/COMST.2017.2671454>.
- Kim, M., Han, D., Rhee, J.K., 2021. Unsupervised view-selective deep learning for practical indoor localization using CSI. *IEEE Sens. J.* <http://dx.doi.org/10.1109/JSEN.2021.3112994>.
- Kim, K.S., Lee, S., Huang, K., 2018. Open access a scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting. pp. 1–17.
- Laska, M., Blankenbach, J., 2021. Deeplocbox: Reliable fingerprinting-based indoor area localization. *Sensors* 21 (6), 1–23. <http://dx.doi.org/10.3390/s21062000>.
- Liu, F., et al., 2020. Survey on WiFi-based indoor positioning techniques. *IET Commun.* 14 (9), 1372–1383. <http://dx.doi.org/10.1049/iet-com.2019.1059>.
- Lohan, E.S., Torres-Sospedra, J., Leppäkoski, H., Richter, P., Peng, Z., Huerta, J., 2017. Wi-Fi crowdsourced fingerprinting dataset for indoor positioning. *Data* 2 (4), 1–16. <http://dx.doi.org/10.3390/data2040032>.
- Lu, X., Zou, H., Zhou, H., Xie, L., Bin Huang, G., 2016. Robust extreme learning machine with its application to indoor positioning. *IEEE Trans. Cybern.* 46 (1), 194–205. <http://dx.doi.org/10.1109/TCYB.2015.2399420>.
- Mendoza-Silva, G.M., Torres-Sospedra, J., Huerta, J., 2019. A meta-review of indoor positioning systems. *Sensors (Switzerland)* 19 (20), <http://dx.doi.org/10.3390/s19204507>.
- Min, M., et al., 2021. 3D geo-indistinguishability for indoor location-based services. *IEEE Trans. Wirel. Commun.* 1–14. <http://dx.doi.org/10.1109/TWC.2021.3132464>.
- Nayak, D.R., Dash, R., Majhi, B., Pachori, R.B., Zhang, Y., 2020. A deep stacked random vector functional link network autoencoder for diagnosis of brain abnormalities and breast cancer. *Biomed. Signal Process. Control* 58, 101860. <http://dx.doi.org/10.1016/j.bspc.2020.101860>.
- Nowicki, M., Wietrzychowski, J., 2017. Low-effort place recognition with WiFi fingerprints using deep learning. In: *Advances in Intelligent Systems and Computing*, Vol. 550. Springer Verlag, pp. 575–584.
- Oh, Y., Noh, H.M., Shin, W., 2021. C-CNNLoc: Constrained CNN for robust indoor localization with building boundary. *Electron. Lett.* 57 (10), 422–425. <http://dx.doi.org/10.1049/ELL2.12142>.
- Pang, S., Yang, X., 2016. Deep convolutional extreme learning machine and its application in handwritten digit classification. *Comput. Intell. Neurosci.* 2016, <http://dx.doi.org/10.1155/2016/3049632>.
- Pao, Y.H., Takefuji, Y., 1992. Functional-link net computing: Theory, system architecture, and functionalities. *Computer (Long. Beach. Calif.)* 25 (5), 76–79. <http://dx.doi.org/10.1109/2.144401>.
- Qin, F., Zuo, T., Wang, X., 2021. CCPOS: WiFi fingerprint indoor positioning system based on cdae-cnn. *Sensors (Switzerland)* 21 (4), 1–17. <http://dx.doi.org/10.3390/s21041114>.
- Raitoharju, M., García-Fernández, F., Hostettler, R., Piché, R., Särkkä, S., 2020. Gaussian mixture models for signal mapping and positioning. *Signal Process.* 168, 107330. <http://dx.doi.org/10.1016/j.sigpro.2019.107330>.
- Sanam, T.F., Godrich, H., 2020. A multi-view discriminant learning approach for indoor localization using amplitude and phase features of CSI. *IEEE Access* 8, 59947–59959. <http://dx.doi.org/10.1109/ACCESS.2020.2982277>.
- Schmidt, W.F., Kraaijveld, M.A., Duin, R.P.W., 1992. Feed forward neural networks with random weights. In: *Proc. - Int. Conf. Pattern Recognit.*, Vol. 2. pp. 1–4. <http://dx.doi.org/10.1109/ICPR.1992.201708>.
- Shi, Q., Katuwal, R., Suganthan, P.N., Tanveer, M., 2021. Random vector functional link neural network based ensemble deep learning. *Pattern Recognit.* 117, 107978. <http://dx.doi.org/10.1016/j.patcog.2021.107978>.
- Singh, N., Choe, S., Punmiya, R., 2021. Machine learning based indoor localization using Wi-Fi RSSI fingerprints: An overview. *IEEE Access* 9, 127150–127174. <http://dx.doi.org/10.1109/ACCESS.2021.3111083>.
- Sinha, R.S., Hwang, S.H., 2019. Comparison of CNN applications for rssi-based fingerprint indoor localization. *Electron.* 8 (9), <http://dx.doi.org/10.3390/electronics8090989>.
- Sithole, G., Zlatanova, S., 2016. Position, location, place and area: An indoor perspective. In: *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, III–4. pp. 89–96. <http://dx.doi.org/10.5194/ISPRS-ANNALS-III-4-89-2016>.
- Song, X., et al., 2019. A novel convolutional neural network based indoor localization framework with WiFi fingerprinting. *IEEE Access* 7, 110698–110709. <http://dx.doi.org/10.1109/access.2019.2933921>.
- Spachos, P., Plataniotis, K.N., 2020. Ble beacons for indoor positioning at an interactive IoT-based smart museum. *IEEE Syst. J.* 14 (3), 3483–3493. <http://dx.doi.org/10.1109/JSYST.2020.2969088>.
- Suganthan, P.N., Katuwal, R., 2021. On the origins of randomization-based feedforward neural networks. *Appl. Soft Comput.* 105, 107239. <http://dx.doi.org/10.1016/j.asoc.2021.107239>.
- Tang, Jiexiang, Deng, C., Huang, Guang-Bin, 2015. Extreme learning machine for multilayer perceptron. *IEEE Trans. Neural Networks Learn. Syst.* 27 (4), 809–821. <http://dx.doi.org/10.1109/TNNLS.2015.2424995>.
- Tiglao, N.M., Alipio, M., Dela Cruz, R., Bokhari, F., Rauf, S., Khan, S.A., 2021. Smartphone-based indoor localization techniques: State-of-the-art and classification. *Measurement* 179, 109349. <http://dx.doi.org/10.1016/j.measurement.2021.109349>.
- Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, Ó., Huerta, J., 2015. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Syst. Appl.* 42 (23), 9263–9278. <http://dx.doi.org/10.1016/j.eswa.2015.08.013>.
- Torres-Sospedra, J., et al., 2014. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In: *IPIN 2014–2014 International Conference on Indoor Positioning and Indoor Navigation*. pp. 261–270. <http://dx.doi.org/10.1109/IPIN.2014.7275492>.
- Uphaus, P.O., Beringer, B., Siemens, K., Ehlers, A., Rau, H., 2021. Location-based services—the market: Success factors and emerging trends from an exploratory approach. *J. Locat. Based Serv.* <http://dx.doi.org/10.1080/17489725.2020.1868587>.
- Wang, Q., 2020. A robust and accurate indoor localization system using deep auto-encoder combined with multi-feature fusion. *J. Ambient Intell. Humaniz. Comput.* (0123456789), <http://dx.doi.org/10.1007/s12652-020-02438-5>.
- Xia, S., Liu, Y., Yuan, G., Zhu, M., Wang, Z., 2017. Indoor fingerprint positioning based on WiFi: An overview. *ISPRS Int. J. Geo-Inform.* 6 (5), <http://dx.doi.org/10.3390/ijgi6050135>.
- Yao, C.Y., Hsia, W.C., 2018. An indoor positioning system based on the dual-channel passive RFID technology. *IEEE Sens. J.* 18 (11), 4654–4663. <http://dx.doi.org/10.1109/JSEN.2018.2828044>.
- Yiu, S., Dashti, M., Claussen, H., Perez-Cruz, F., 2017. Wireless RSSI fingerprinting localization. *Signal Process.* 131, 235–244. <http://dx.doi.org/10.1016/j.sigpro.2016.07.005>.
- Zeng, Y., Xu, X., Fang, Y., Zhao, K., 2015. Traffic sign recognition using deep convolutional networks and extreme learning machine. In: *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9242, pp. 272–280. http://dx.doi.org/10.1007/978-3-319-23989-7_28_COVER/.
- Zhang, L., Suganthan, P.N., 2016. A survey of randomized algorithms for training neural networks. *Inf. Sci. (Ny)* 364–365, 146–155. <http://dx.doi.org/10.1016/j.ins.2016.01.039>.
- Zhang, L., Suganthan, P.N., 2017. Visual tracking with convolutional random vector functional link network. *IEEE Trans. Cybern.* 47 (10), 3243–3253. <http://dx.doi.org/10.1109/TCYB.2016.2588526>.
- Zhu, X., Qu, W., Qiu, T., Zhao, L., Atiquzzaman, M., Wu, D.O., 2020. Indoor intelligent fingerprint-based localization: Principles, approaches and challenges. *IEEE Commun. Surv. Tutor.* (c), 1. <http://dx.doi.org/10.1109/comst.2020.3014304>.