sensors

MDPI

Article

Calibration-Free 3D Indoor Positioning Algorithms Based on DNN and DIFF

Jingmin Yang [1,2,3,*], Shanghui Deng [1,2], Li Xu [3] and Wenjie Zhang [1,2]

[1] School of Computer Science, Minnan Normal University, Zhangzhou 363000, China
[2] Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou 363000, China
[3] Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350007, China
* Correspondence: yjm1758@mnnu.edu.cn

Abstract: The heterogeneity of wireless receiving devices, co-channel interference, and multi-path effect make the received signal strength indication (RSSI) of Wi-Fi fluctuate greatly, which seriously degrades the RSSI-based positioning accuracy. Signal strength difference (DIFF), a calibration-free solution for handling the received signal strength variance between diverse devices, can effectively reduce the negative impact of signal fluctuation. However, DIFF also leads to the explosion of the RSSI data dimension, expanding the number of dimensions from m to C_m^2, which reduces the positioning efficiency. To this end, we design a data hierarchical processing strategy based on a building-floor-specific location, which effectively improves the efficiency of high-dimensional data processing. Moreover, based on a deep neural network (DNN), we design three different positioning algorithms for multi-building, multi-floor, and specific-location respectively, extending the indoor positioning from the single plane to three dimensions. Specifically, in the stage of data preprocessing, we first create the original RSSI database. Next, we create the optimized RSSI database by identifying and deleting the unavailable data in the RSSI database. Finally, we perform DIFF processing on the optimized RSSI database to create the DIFF database. In the stage of positioning, firstly, we design an improved multi-building positioning algorithm based on a denoising autoencoder (DAE). Secondly, we design an enhanced DNN for multi-floor positioning. Finally, the newly deep denoising autoencoder (DDAE) used for specific location positioning is proposed. The experimental results show that the proposed algorithms have better positioning efficiency and accuracy compared with the traditional machine learning algorithms and the current advanced deep learning algorithms.

Keywords: calibration-free; deep denoising autoencoder (DDAE); fingerprint database; signal strength difference (DIFF); 3D indoor positioning

check for updates

Citation: Yang, J.; Deng, S.; Xu, L; Zhang, W. Calibration-Free 3D Indoor Positioning Algorithms Based on DNN and DIFF. *Sensors* **2022**, *22*, 5891. https://doi.org/10.3390/s22155891

Academic Editors: Binghao Li and Chris Rizos

Received: 29 June 2022
Accepted: 5 August 2022
Published: 7 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

With the widespread popularity of mobile terminal devices and the rapid development of Internet technologies, smart devices play an increasingly important role in the field of smart cities, providing people with various smart services, such as smart homes, smart healthcare, smart transportation systems, etc. [1–3]. With the wide application of GPS, GLONASS, BDS, GALILEO, and other global positioning systems, outdoor location-based services (LBS) play an important role in people's daily life, such as emergency services, employee management, and vehicle tracking [4,5]. In recent years, with the rapid growth of mobile devices, indoor positioning has become more and more important, such as the indication of vacant parking spaces in indoor parking lots, the location monitoring of infants in hospitals, and the route guidance for customers in large shopping malls, etc. [6–8]. However, indoor positioning still faces the challenge of low positioning accuracy. Therefore, indoor positioning is still an issue worth studying.

The common technologies for indoor positioning include radio frequency identification device (RFID) [9,10], ZigBee [11,12], bluetooth [13], ultra-wideband (UWB) [14], Wi-Fi [15], etc. RFID technology has low power consumption and a wide range of applications, but it is prone to interference, resulting in low positioning accuracy. ZigBee is widely used in wireless sensor networks. But this technology is not available on commonly used devices, such as mobile phones. Therefore, additional equipment is required, which increases extra costs. Bluetooth has the advantages of high throughput, wide receiving range, and low energy consumption. But it is prone to noise, resulting in low positioning accuracy. UWB has high accuracy with strong anti-interference to multipath effects, but has a relatively short range and requires additional hardware support, resulting in high costs. A wireless local area network (WLAN) is developed based on the IEEE 802.11 series standards issued by the Institute of Electrical and Electronics Engineers (IEEE). It uses high-frequency wireless signals as the transmission medium, such as wireless electromagnetic waves in the 2.4 GHz or 5 GHz band. In 2018, a textbook titled "Wireless network technology: principle, experiment, and network design", written by us, states that WLAN has the characteristics of flexible use, convenient expansion, effective costs, and simple installation [16]. Meanwhile, with the advancement of wireless communication technologies and the development of indoor communication requirements, WLAN is developing rapidly. On the other hand, compared with the aforementioned indoor positioning technologies, Wi-Fi-supported devices are diverse and inexpensive, and Wi-Fi signals are easy to be measured. These advantages make Wi-Fi networks widely used for indoor positioning [17].

The traditional indoor positioning methods such as angle of arrival (AoA), time of arrival (ToA), and time difference of arrival (TDoA) all require line-of-sight (LoS) measurements. The process of collecting fingerprint signals and correlating indoor locations without measuring LoS has become one of the prevalent research methods in the field of indoor positioning [18]. Fingerprint-based positioning methods are usually divided into two stages, the offline and the online [19]. In the offline phase, the RSSI of different reference locations is precisely measured, and a fingerprint database with RSSI and location mapping is created. In the online phase, specific coordinates are calculated by comparing the RSSI measured at the point to be located with the fingerprint database obtained in the offline phase.

In most cases, the RSSI-based Wi-Fi fingerprint positioning methods have high feasibility [20]. However, the current RSSI-based Wi-Fi fingerprint positioning methods still have the following issues that affect the positioning accuracy: (1) When the Wi-Fi signal is blocked by buildings, walls, and other objects during the propagation process, the multipath effect is more likely to occur. (2) With the widespread popularity of mobile terminals, The heterogeneity of wireless devices leads to large fluctuations in the RSSI value of the same anchor point received at the same position. (3) At present, the commonly used Wi-Fi frequency band is 2.4 GHz. When multiple devices work at the same time, the co-channel interference increases the noise of the RSSI value received by the devices. We analyze these issues in detail in a study published in 2020 and note that these issues severely degrade RSSI-based positioning accuracy [21].

To address the above issues, we propose calibration-free indoor positioning algorithms based on DNN and DIFF. Compared with the existing Wi-Fi fingerprint database positioning methods, the main contributions of our work are summarized as follows.

(1) Aiming at the issue of positioning inefficiency caused by the data dimension explosion of DIFF, we design a data hierarchical processing strategy based on building-floor-specific location, which effectively improves the efficiency of high-dimensional data processing. Correspondingly, we create the original RSSI database, the optimized RSSI database, and the DIFF database. The optimized RSSI database is for multi-building positioning, and the DIFF database is for multi-floor and specific-location positioning. The update and processing of these three databases are performed in an asynchronous mode.

(2) The proposed DNN-based algorithms extend the indoor location from a single plane to three dimensions, namely, multi-building, multi-floor and specific locations. We de-

sign an improved DAE algorithm based on optimized RSSI for multi-building positioning, which improves the positioning accuracy up to 100%. On this basis, We design an enhanced DNN algorithm for multi-floor positioning. Finally, a newly designed specific-location positioning algorithm called DDAE is proposed. The DDAE consists of an optimized DAE and an enhanced DNN. The key idea is to utilize the DAE to remove redundant information and reduce the dimensions of input data DIFF while retaining the key features. Then, the dimensionality-reduced features are used as the input for the next part of the enhanced DNN. We evaluate our algorithms on the UJIIndoorLoc dataset compared with various baseline machine learning algorithms and deep learning algorithms. The results show that the proposed algorithms have better positioning efficiency, and the average positioning error at the specific position is also the smallest without extra calibration.

The rest of this paper is organized as follows. Section 2 reviews the related work of indoor positioning. Section 3 introduces the 3D indoor positioning algorithms. Section 4 presents the experimental preparation, including the RSSI-based data set and its optimization, the processing of DIFF, the establishment of the calibration-free DIFF database, and the evaluation indicators for algorithms. Section 5 studies the optimization of the algorithms, as well as evaluates the performance of the proposed algorithms. Finally, Section 6 summarizes the whole study work and points out the future research work.

2. Related Work

In general, according to whether the distance is measured, the indoor positioning methods can be divided into two types:ranging-based and non-ranging-based [22]. The ranging-based positioning methods have been widely used in outdoor environments, such as ToA, TDoA, AoA, etc. Meanwhile, RSSI-based fingerprint positioning is a commonly used indoor positioning method without ranging [23]. This method includes two stages: the offline fingerprint collection stage and the online fingerprint positioning stage. The purpose of the offline collection fingerprint stage is to build a database that matches specific location coordinates and RSSI fingerprint features. Specifically, at each collection point, by collecting the RSSI values of different access points (APs), the corresponding relationship between the RSSI features and the coordinates of the collection point can be established. In the online positioning phase, the RSSI of the to-be-located point is collected as the input data of the positioning algorithm, and the specific coordinates of the to-be-located point are calculated by the positioning algorithm.

In specific practice, the heterogeneity of wireless receiving devices, co-channel interference, and multi-path effect make the RSSI of Wi-Fi fluctuate greatly, which seriously degrades the RSSI-based positioning accuracy and robustness [21]. At present, there are two main methods to solve the problem of RSSI fingerprint fluctuation: calibration methods and calibration-free methods. The calibration methods establish the RSSI relationship model of the reference devices and the to-be-located mobile terminal at the same location in the offline stage. In the online phase, the RSSI relationship model is used to compensate for the RSSI measured of the to-be-located mobile terminal. The calibration methods have obtained many research results. The most significant advantage of calibration methods is that it is straightforward and effective. However, these methods also have some problems, such as higher delay in the calibration process [24]. The calibration-free methods optimize the original RSSI fingerprint to obtain a new fingerprint with better robustness. For example, the hyperbolic location fingerprinting (HLF) method proposed by Kjaergaard [25] uses the pairwise ratio of the collected RSSI as new fingerprints. The HLF achieves better localization accuracy but also increases the computational complexity of fingerprint comparison. In the reference [26], a calibration-free method of fingerprints called DIFF is proposed. The DIFF constructs a new fingerprint database by extracting the pairwise signal strength differences from the raw RSSI values. Through experimental comparison, the performance of the DIFF is better than that of the HLF.

A large number of RSSI-based machine learning methods have been proposed for indoor positioning,such as support vector machines (SVM) [27], k-nearest neighbors

(KNN) [28], random forests (RF) [29], decision trees (DT) [30], and Gaussian Naïve-Bayes (GNB) [31]. In recent years, with the rapid development of deep learning, more and more deep learning technologies are applied to the field of indoor positioning. In 2017, Nowicki et al. designed a SAEDNN based on stacked autoencoder (SAE) and DNN for floor positioning [32]. In 2018, Jang et al. propose a CNN-based classifier for multi-building positioning [33]. These experiments are conducted on the UJIIndoor dataset [28]. In 2018, Kim et al. propose an improved model called SAEDNN, which reduces the number of nodes in the last layer [34]. In 2019, Song et al. propose an improved model called CNNLoc, which is proved to be superior to the above algorithms [35]. However, the above methods do not take into account the positioning of specific locations within a given floor, which is a more challenging issue. In 2021, Qin et al. propose a CDAE-CNN-based Wi-Fi fingerprint location algorithm called Ccpos, which uses the K-means algorithm to segment the dataset, and then uses the CDAE-CNN network for specific location prediction [36]. In 2022, Wang et al. designed a novel DNN-based indoor positioning framework called CHISEL, which combines a convolutional encoder and a CNN classifier [37]. Jaehoon Cha et al. propose a hierarchical auxiliary deep neural network called HADNN, which uses a continuous feedforward network to identify buildings and estimate the floor coordinates [38].

The above research work has achieved amazing findings, but the influence of RSSI fluctuation is not considered separately. In other words, the above work unconsciously deals with this problem by using the technologies of deep learning. Inspired by the research work of the above-mentioned outstanding scholars, we combine the DIFF and deep learning to design new algorithms that extend indoor localization from plane positioning to 3D positioning. In the following sections, the key idea and specific design of these algorithms are discussed in detail.

3. Methodology

This section first introduces the overall structural design of the positioning system and then presents the theoretical basis and the design of the multi-building positioning model, multi-floor positioning model, and specific-location positioning model in detail.

3.1. Positioning System Architecture

The overall positioning system architecture proposed in this paper is shown in Figure 1.

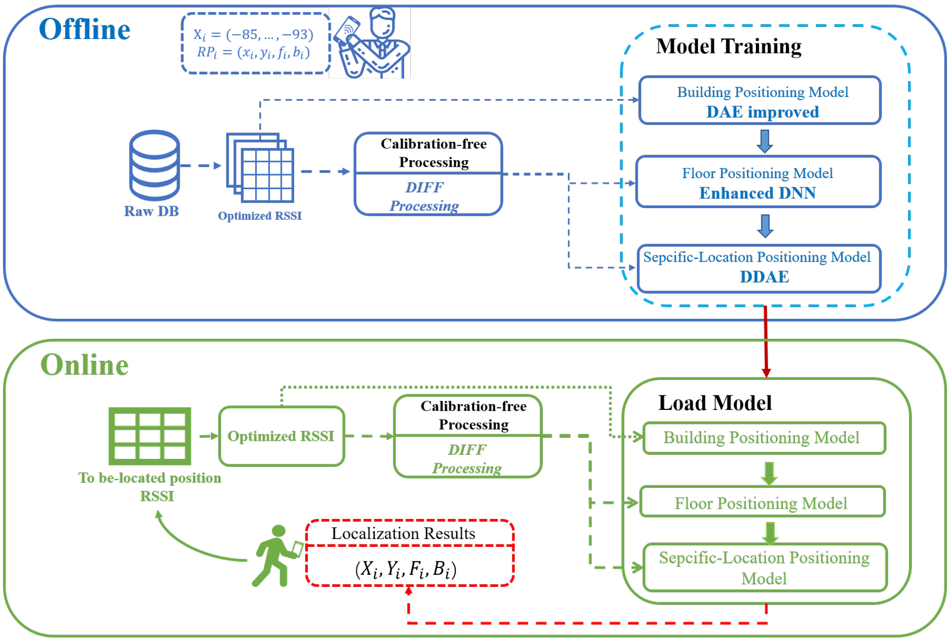

Offline
$X_i = (-85, \ldots, -93)$
$RP_i = (x_i, y_i, f_i, b_i)$
Model Training
Building Positioning Model
DAE improved
Raw DB
Optimized RSSI
Calibration-free Processing
DIFF Processing
Floor Positioning Model
Enhanced DNN
Sepcific-Location Positioning Model
DDAE
Online
Load Model
To be-located position RSSI
Optimized RSSI
Calibration-free Processing
DIFF Processing
Building Positioning Model
Floor Positioning Model
Sepcific-Location Positioning Model
Localization Results
(X_i, Y_i, F_i, B_i)

Figure 1. System Architecture.

The indoor positioning process is mainly composed of the offline training phase and the online positioning phase. In the offline training phase, firstly, we collect RSSI and build the original RSSI fingerprint database. Secondly, we identify and delete the unavailable data in the RSSI database, and then create the calibration-free DIFF fingerprint database based on the optimized RSSI database. The update of the RSSI database, the optimization of RSSI data, and the update of the DIFF database adopt asynchronous operation mode and have no impact on each other. Finally, we use the optimized RSSI database to train the multi-building positioning model, and the DIFF database to train the multi-floor positioning model and specific-location positioning model, respectively. In the online positioning stage, the RSSI fingerprint measured at the to-be-located position is optimized first, and then converted into the calibration-free DIFF. Then, through the trained positioning algorithms, the 3D coordinate values of the building-floor-specific location are calculated, respectively, and returned to the requester.

3.2. Multi-Building Positioning Model

Autoencoder (AE) is a kind of unsupervised learning artificial neural network (ANN). Its main function is to perform representation learning on the input information, remove redundant information, and retain the key features. AE is widely used in dimensionality reduction and anomaly detection [39]. AE consists of the encoder and the decoder. Given high-dimensional input space X, the low-dimensional feature space F is obtained through the encoder f, and then the high-dimensional space $\hat{X}$ is restored through the decoder g. The AE solves the mapping f and g by minimizing the reconstruction error of the input feature X and the restored feature $\hat{X}$, that is,

$$f : X \to F \tag{1}$$

$$g : F \to \hat{X} \tag{2}$$

$$f, g = \underset{f,g}{argmin} ||X - g[f(X)]||^2 \tag{3}$$

Affected by problems such as model complexity, the training set data volume, and data noise, the initial model obtained through AE often has the risk of over-fitting. The DAE solves this problem effectively by reconstructing the input data with noise [40]. The DAE model is shown in Figure 2.

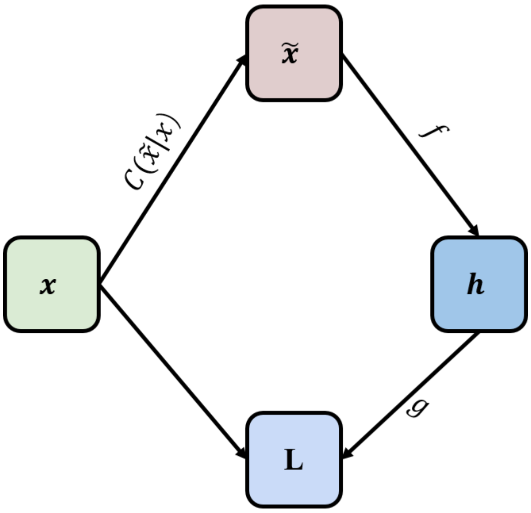


Figure 2. DAE Network.

Figure 2 introduces a noise process $C(\tilde{x}|x)$. $C(\tilde{x}|x)$ is a conditional probability, which represents the probability of generating noise samples $\tilde{x}$ given the data sample x. DAE is trained to reconstruct the original sample x from the noise-added $\tilde{x}$ by minimizing the loss function $L = -logp_{decoder}(x|h = f(\tilde{x}))$, where $\tilde{x}$ is the sample obtained from the initial sample x after the noise processing $C(\tilde{x}|x)$. Usually, the distribution function $p_{decoder}$ is the distribution of the DAE neural network parameters. DAE learns the reconstruction distribution $p_{decoder}$ from the training data pair $(x, \tilde{x})$ according to the following process:

1. Randomly take a training sample x from the training set.
2. Randomly take a noise sample $\tilde{x}$ from the noise process $C(\tilde{x}|x)$.
3. Estimate the reconstruction distribution $p_{decoder}(x|\tilde{x}) = p_{decoder}(x|h)$ of the ADE by using the data pair $(x, \tilde{x})$ as training samples.

In general, we can use the gradient descent method to minimize the approximate solution of the negative log-likelihood function $-logp_{decoder}(x|h)$. Therefore, DAE can be expressed as a random gradient descent function under the following expectations, where $\hat{p}_{data}$ is the distribution of the training data:

$$-E_{x\sim\hat{p}_{data}(x)}E_{(\tilde{x}\sim C)}(\tilde{x}|x)logp_{decoder}(x|h = f(\tilde{x})) \tag{4}$$

When adopting the hierarchical positioning strategy, the classification accuracy of multi-building is the first key issue. Because the wrong classification of the building makes subsequent multi-floor localization and specific location localization meaningless. After analysis, we find that the distance between buildings is relatively long, and the difference in Wi-Fi signals in each building is obvious. Therefore, we use the optimized RSSI as the input value of the model and design an improved DAE to Increase the anti-noise ability of the model. After the training of DAE, the denoised features are connected to a fully connected hidden layer, and finally the buildings are classified by the softmax layer. The processing process is shown in Figure 3.

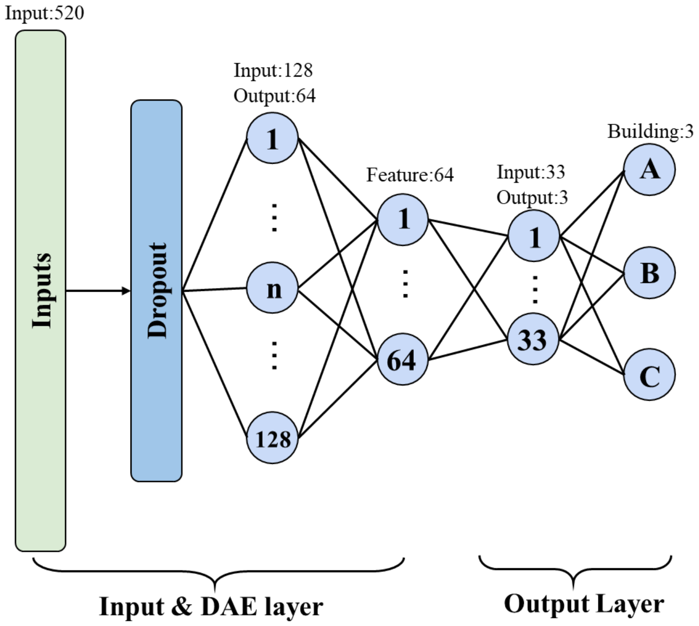


Figure 3. Multi-building positioning Model.

3.3. Multi-Floor Positioning Model

In the multi-floor positioning problem, we use an enhanced DNN as the floor positioning model, as shown in Figure 4.

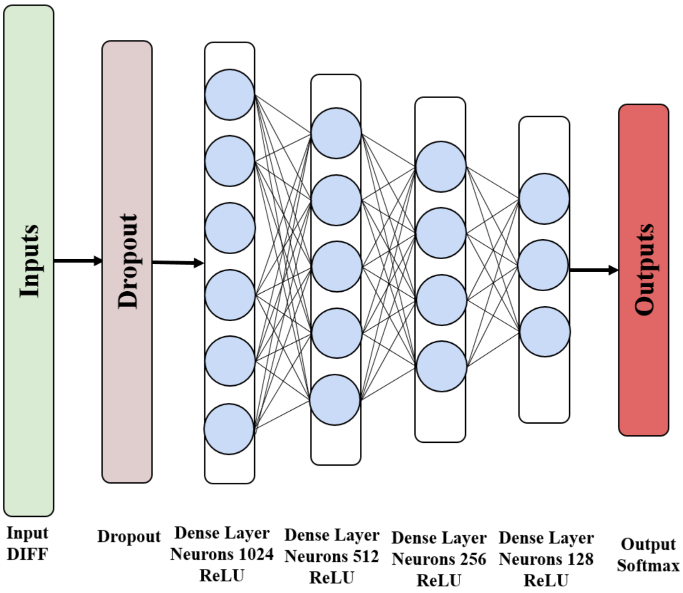

Inputs
Dropout
Outputs
Input DIFF
Dropout
Dense Layer Neurons 1024 ReLU
Dense Layer Neurons 512 ReLU
Dense Layer Neurons 256 ReLU
Dense Layer Neurons 128 ReLU
Output Softmax

Figure 4. DNN classification model.

The enhanced DNN model takes DIFF as input data and is trained by the stochastic gradient descent (SGD) algorithm. SGD uses the error back-propagation (BP) algorithm to calculate the loss function between the model prediction and the truth value of the floor. The BP algorithm is composed of two processes, the forward propagation of the signal and the back-propagation of the error. In the forward propagation stage, the DIFF samples enter the DNN from the input layer and are transferred layer by layer to the output layer. If the error between the actual output of the DNN and the expected output is too large, the algorithm turns to the error back-propagation. In the error back-propagation stage, the BP algorithm reversely calculates the error in the DNN, and continuously adjusts the parameters of the neurons θ (weight and bias) in each layer during the back-propagation process to minimize the error. To avoid the disappearance of the gradient, the hidden layer uses the ReLU function as the activation function, and the output layer uses the softmax function defined in (5) to map the input data DIFF to the prediction probability of the real floor to complete the floor classification prediction.

$$\sigma(z_j) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, j = 1, \ldots, K \tag{5}$$

where all the z_j values are the elements of the input vector, and any real value can be taken. The term at the bottom of the Formula (5) is the normalization term, which ensures that the sum of all output values of the function is 1, thus constituting a valid probability distribution.

The improved DNN chooses cross-entropy as the loss function. In addition, L_2 regularization is used to avoid over-fitting. Therefore, the learning of DNN is to minimize the loss function of the parameter set θ in the loss function $L(t, y, \theta)$ defined by Formula (6).

$$L(t, y, \theta) = -\sum_{s=1}^{K} t_s log y_s + \lambda \sum_i \theta_i^2 \tag{6}$$

where t and y represent the target output and the predicted output vectors of the multi-layer perceptron (MLP). The cross-entropy indicates the difference between the amount of information contained in t and y. If the value of the cross-entropy is smaller, then the predicted output is closer to the target output. The second term of the Formula (6) is the regular term $\lambda \sum_i \theta_i^2$, where λ is the regularization parameter that needs to be optimized, and the value range is [0, $+\infty$), and $\sum_i \theta_i^2$ represents the sum of squares of the weight parameters.

3.4. Sepcific-Location Positioning Model

After completing the positioning of the building and floor of the mobile user, we need to further locate the specific location of the mobile user. To this end, we design a new DDAE model, as shown in Figure 5. The DDAE model consists of two parts: improved DAE and enhanced DNN. The improved DAE increases the number of hidden layers to 4. At the same time, during the model training process, an improved gradient descent method called adaptive moment estimation (Adam) is used to update the parameter set θ (weights and biases) of the model. Algorithm 1 gives the pseudo-code of DDAE offline training. The Input includes the DIFF value of all training sets, location labels, max_epoch, dropout rate, and learning rate. Firstly, the algorithm initializes all weights and biases randomly (step 1). Then for each epoch, the algorithm randomly selects a mini_batch from the DIFF values of all training sets and feeds these data to the DDAE (step 3). The first and last layers of the DDAE are the input layer and the output layer, respectively. Starting from the second layer, the input data is processed by the dropout layer and the hidden layers (steps 6–17). Step 19 uses the loss function to measure the error between the true position of the training set and the output value of DDAE.

Algorithm 1 DDAE Weight Training Algorithm

```
Input: DIFF value, network architecture, max_epoch, dropout_rate $p$, and learning rate $\alpha$;
Output: Trained weights $\omega$ and $b$;
 1: Randomly initialize $\omega$ and $b$;
 2: while $epoch < Max\_epoch$ do
 3:   Randomly select a mini-batch from inputs;
 4:   // Forward propagation;
 5:   // $L$ is the number of layers of the DDAE;
 6:   for $l$ = 2:L-2 do
 7:     if the current layer is a dropout layer then
 8:       $r_j^{(l)} \sim Bernoulli(p)$
 9:       $\tilde{y}^{(l)} = r^{(l)} \cdot y^{(l)}$
10:       $z_i^{(l+1)} = \omega_i^{(l+1)}\tilde{y}^{(l)} + b_i^{(l+1)}$
11:       $y_i^{(l+1)} = f(z_i^{(l+1)})$
12:     else
13:       // The current layer is a hidden layer;
14:       $z^{(l)} = \omega^{(l)}y^{(l)} + b^{(l)}$
15:       $y^{(l)} = f_l(z^{(l)})$
16:     end if
17:   end for
18:   //Loss function;
19:   $Loss = -\sum_{i=1}^{output\_size} y_i \cdot log\hat{y}_i$
20: end while
```

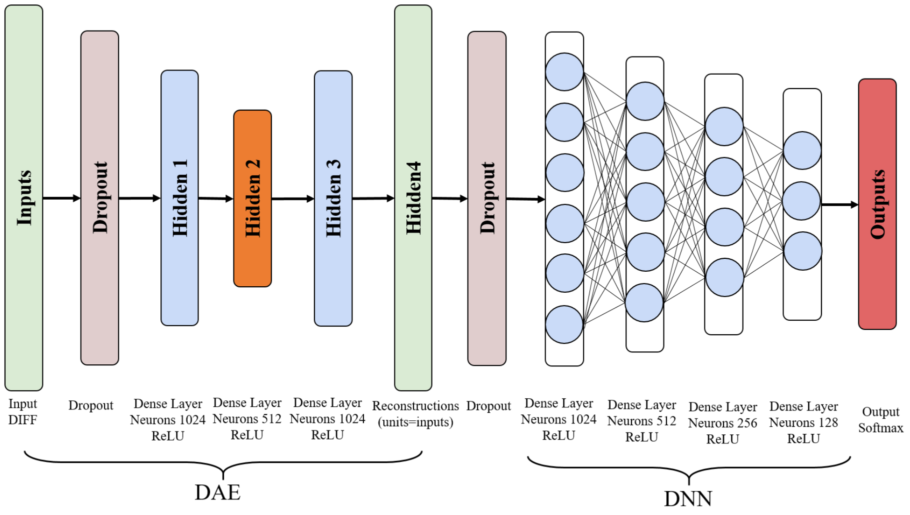

Inputs
Dropout
Hidden 1
Hidden 2
Hidden 3
Hidden4
Dropout
Outputs
Input DIFF
Dropout
Dense Layer Neurons 1024 ReLU
Dense Layer Neurons 512 ReLU
Dense Layer Neurons 1024 ReLU
Reconstructions (units=inputs)
Dropout
Dense Layer Neurons 1024 ReLU
Dense Layer Neurons 512 ReLU
Dense Layer Neurons 256 ReLU
Dense Layer Neurons 128 ReLU
Output Softmax
DAE
DNN

Figure 5. DDAE Location Estimation Model.

4. Experimental Setup

This section mainly introduces the UJIIndoorLoc data set used in the experiment and describes in detail the environment in which the data set is collected, as well as the attributes and characteristics contained in this data set. Next, it introduces the core process of data preprocessing and calibration-free processing. Finally, the evaluation indexes used in the experiment are presented.

4.1. UJIIndoorLoc Dataset

In the literature related to indoor positioning, UJIIndoorLoc is the comprehensive and accessible data set in the UCI machine learning repository [28]. This data set contains the Wi-Fi RSSI data used in the IPIN 2015 EvAAL competition and is publicly available [41]. The collected data comes from 25 different Android devices with the help of 20 volunteers. Specifically, the data set consists of 21,048 Wi-Fi RSSI data, which are divided into 19,937 training samples and 1,111 verification samples. These samples are collected in the buildings of the University of Jaime I in Spain, covering an area of nearly 108,703 square meters. Each collection in the data set contains 529 attributes. In the data collection area, 520 distinct APs are found, so the first 520 attributes are expressed as the RSSI received from these APs. The signal strength ranges from -104 dBm to 0 dBm. When an AP is unavailable, its signal strength value is 100. The remaining 9 attributes are the latitude and longitude of the measurement point, floor number, building ID, space ID, relative location, volunteer ID, phone ID, and the measurement time stamp.

4.2. RSSI Database Optimization

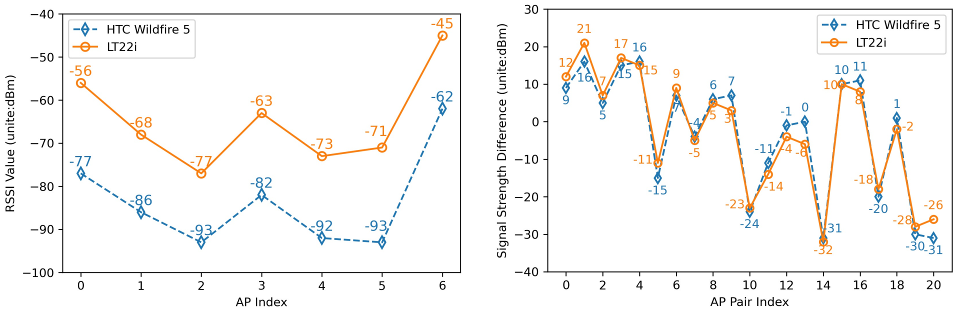
RSSI needs to be optimized before DIFF calibration-free processing to improve positioning accuracy. After analysis, it is found that when the AP is unavailable, its RSSI value is marked as 100, while the available RSSI values range from -104 dBm to 0 dBm. The unavailable APs are identified and deleted. At the same time, the APs are sorted according to the strength of their RSSI, and then the RSSI with stronger signal strength is preferentially selected for positioning.

DIFF brings an explosion of RSSI data dimensions, expanding the dimension size from m to C_m^2, which reduces the positioning efficiency. For example, if the 520 optimized RSSI data are directly processed by DIFF, the data dimension will be extended to 134,940. When the positioning algorithm reads the high-dimensional DIFF data, there will be a problem of insufficient memory, resulting in the low running efficiency of the algorithm. In the following experimental section, we analyze this issue in detail. To this end, we design a data hierarchical processing strategy. Specifically, on the basis of the above RSSI optimization process, we first subdivide the DIFF data set into three sub-datasets according to buildings.

When positioning, the positioning sequence is from multi-building to multi-floor, and then to the specific location. The experimental results show that this is an effective treatment strategy for high-dimensional input data.

4.3. DIFF Calibration-Free Processing

The heterogeneity of wireless receiving devices, co-channel interference, and multi-path effects make the RSSI of Wi-Fi fluctuate greatly. For example, Figure 6a shows the variation curves of RSSI values measured by two different devices at the same location randomly extracted from the RSSI optimization data set.



(**a**) (**b**)

Figure 6. DIFF Calibration-free Processing. (**a**) Signal strength detected by HTC Wildfire 5 and LT22i smartphones at a fixed location. (**b**) Extract signal strength difference from RSSI.

As shown in Figure 6a, the trends of the two curves are the same, but the values are quite different. According to the logarithmic distance loss model, the value of RSSI is inversely proportional to the distance. Therefore, at the fixed position, the RSSI values of different devices all reflect the distance from the fixed point to the AP [21]. On the other hand, by analyzing the shape of the two curves, it is found that the changing trend of the curves can be represented by the RSSI difference between APs. Therefore, we can extract the RSSI difference of AP pairs to replace the original RSSI value [26].

Assuming that there are m APs $R = r_1, r_2, \ldots, r_m$, the RSSI value of the APs received at a specific location can be defined as a row vector $s = (s_1, s_2, \ldots, s_m)$, where s_i represents the RSSI value received from AP r_i. Define the DIFF of an unique AP pair $b_i \times b_j \in R \times R$, and constrain $i < j$ to be unique. The DIFF d can be expressed as:

$$d(r_i, r_j) = s_i - s_j \quad 1 \leq i < j \leq m \tag{7}$$

Therefore, the obtained DIFF feature vector D is expressed as:

$$D = (d(r_1, r_2), d(r_1, r_3), \ldots, d(r_{m-1}, r_m)) \tag{8}$$

where the length of D is t, $t = C_m^2$.

For example, in Figure 6a,

$$S_{HTCWildfire5} = (-77, -86, -93, -82, -92, -93, -62)$$
$$S_{LTzzi} = (-56, -68, -77, -63, -73, -71, -45)$$

so,

$$D_{HTCWildfire5} = (9, 16, 5, 15, 16, -15, 7, -4, 6, 7, -24,$$
$$-11, -1, 0, -31, 10, 11, -20, 1, -30, -31)$$
$$D_{LTzzi} = (12, 21, 7, 17, 15, -11, 9, -5, 5, 3, -23, -14, -4, -6, -32, 10, 8, -18, -2, -28, -26)$$

As shown in Figure 6b, the difference in features extracted by the two different devices is much more similar than the RSSI value.

4.4. Evaluation Index

In multi-building and multi-floor positioning, we use target hit accuracy to evaluate the efficiency of algorithms. In location-specific positioning, we use the average positioning error to evaluate the positioning accuracy of the model. The average positioning error is based on the root mean square error (RMSE), which is defined in the calculation Formula (9) as follows.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_n - \tilde{x_n})^2 + (y_n - \tilde{y_n})^2} \tag{9}$$

where (x_n, y_n) represents the real coordinate location of the positioning user, $(\tilde{x_n}, \tilde{y_n})$ represents the location coordinates calculated by the positioning algorithm, and N represents the number of the calculated positioning coordinates.

5. Performance Evaluation

In this section, we evaluate the proposed positioning algorithms by comparing their performance with existing positioning methods. All experiments in this paper are implemented by programming software under the 64-bit Windows 10 operating system. The main computer hardware parameters are 16 GB memory, Intel Core i5-9500 processor with 3.00 GHz, and NVIDIA GeForce GTX 1660 Ti computing card. We use Python-3.6.2 to program software, and train the deep learning model based on Keras-2.3.1 and Tensorflow-GPU-1.14.0.

5.1. Model Optimization

We optimize the positioning algorithms through multiple experiments. The final optimized parameters are shown in Table 1. The activation functions in the DAE and DNN algorithms are both corrected linear units (ReLU). The optimizer is Adam with a learning rate of 0.001. The loss function is categorical_crossentropy. The activation function of the output layer is Softmax, and the batch training set is 500.

Table 1. Model parameter settings.

Parameter	Value
DAE activation function	ReLU
DAE Optimizer	Adam (lr = 0.001)
DAE loss	categorical_crossentropy
MLP activation function	ReLU
MLP Optimizer	Adam (lr = 0.001)
MLP loss	categorical_crossentropy
Output layer activation function	Softmax
Batch_size	500

The choice and setting of the optimizer is a key task in training a DNN. To obtain the optimal network model, for the evaluation indexes in Section 4.4, we evaluate different optimizers and learning rate combinations through experiments. Evaluated optimizers include Adam, Nadam, RMSprop, and AdaMax [42]. The experimental results are shown in Table 2.

Table 2. Positioning performance of different learning rates in multi-building, multi-floor, and location estimation.

Optimizer	Learning Rate/%	Building Hit Rate/%	Floor Hit Rate/%	Positioning Error /m
Adam	0.005	**100**	96.13	9.80
	0.001	99.91	**96.22**	**6.01**
	0.0005	99.73	95.95	6.60
	0.0001	99.91	96.13	7.61
Nadam	0.005	99.91	95.05	26.49
	0.001	99.64	96.13	6.47
	0.0005	99.46	95.95	6.31
	0.0001	99.73	95.59	8.63
RMSpro	0.005	98.47	95.05	49.73
	0.001	98.11	95.68	7.37
	0.0005	98.20	95.68	7.06
	0.0001	99.55	95.86	6.60
AdaMax	0.05	99.55	82.63	51.59
	0.01	98.83	95.77	7.47
	0.005	99.37	96.13	7.07
	0.001	99.19	95.50	6.83

The experimental results show that the Adam optimizer has outstanding performance. Furthermore, combined with the optimization of the learning rate, optimal positioning results can be obtained. When the learning rate of the Adam optimizer is 0.005, in the multi-building environment, the building hit rate is 100%. When the learning rate of the Adam optimizer is 0.001, the floor hit rate reaches 96.22%. When the learning rate of the Adam optimizer is 0.001, the positioning error of the positioning estimation is the smallest, which is reduced to 6.01 m.

5.2. Performance Analysis of Building and Floor Positioning Models

To evaluate the positioning accuracy of the proposed algorithms in multi-building and multi-floor scenarios, we compare them with traditional machine learning algorithms and current advanced deep learning algorithms, respectively. The experimental results show that our algorithms have obvious advantages in multi-floor and multi-floor positioning scenarios, whether compared with traditional machine learning algorithms or deep learning algorithms. The specific analysis is as follows.

Machine learning algorithms used for comparison include SVM [27], KNN [28], RF [29], DT [30], and GNB [31]. The experimental results are shown in Table 3.

Table 3. Comparison of our algorithm and machine learning methods for multi-building positioning.

Multi-Building Positioning Model	DT	GNB	KNN	SVM	RF	Our
positioning accuracy (%)	97.03	99.19	99.46	99.72	99.91	100

It can be seen from Table 3 that the hit rate of machine learning methods in building positioning is 97.03% at the lowest (DT) and 99.91% at the highest (RF), while the DAE model proposed in this paper reaches 100%. This indicates that the improved DAE model can effectively handle the multi-building positioning issue. Figure 7 shows the result of multi-floor positioning. As can be seen from Figure 7, in terms of multi-floor positioning, the hit rate of machine learning methods is 40.41% at the lowest (GNB), and 92.44% at the highest (SVM), while the DNN model proposed in this paper reaches 96.22%. The enhanced DNN model greatly improves the multi-floor positioning accuracy.

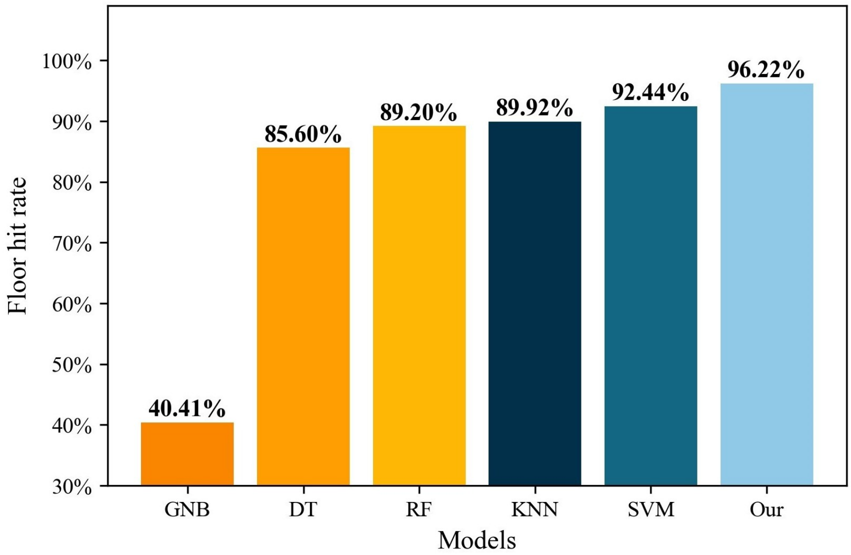


Figure 7. Comparison of our algorithm and machine learning methods for multi-floor positioning.

To further evaluate the performance of our improved DAE and enhanced DNN algorithms, we compare them with state-of-the-art deep learning indoor positioning algorithms, including SAEDNN [34], CNNLoc [35], CCpos [36], CHISEL [37], CHISEL-DA [37], and HADNN [38]. The results are shown in Table 4 and Figure 8, respectively.

Table 4. Comparison of our algorithm and state-of-the-art deep learning methods for multi-building positioning.

Multi-Building Positioning Model	CNNLoc	CCpos	CHISEL	SAEDNN	CHISEL-DA	HADNN	Our
positioning accuracy (%)	96.03	99.60	99.64	99.82	99.96	100.00	100.00

Table 4 shows the results of the above-mentioned indoor positioning technologies in multi-building positioning. It can be seen from Table 4 that the hit rate of deep learning methods in building positioning is 96.03% at the lowest (CNNLoc) and 100% at the highest (HADNN), meanwhile, the DAE model proposed in this paper reaches 100%. This indicates that the improved DAE model can effectively handle the multi-building positioning issue.

Figure 8 shows the results of the above-mentioned deep learning technologies in multi-floor positioning. As can be seen from Figure 8, in terms of multi-floor positioning, the hit rate of deep learning methods is 91.27% at the lowest (SAEDNN), and 95.30% at the highest (CCpos), while the enhanced DNN model proposed in this paper reaches 96.22%. The enhanced DNN model greatly improves the multi-floor positioning accuracy.

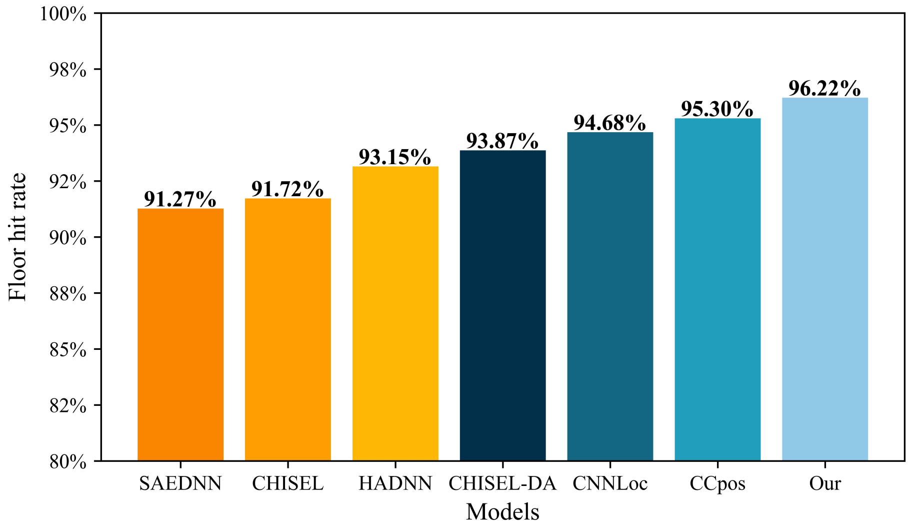


Figure 8. Comparison of our algorithm and state-of-the-art deep learning methods for multi-floor positioning.

5.3. Performance Analysis of Specific-Location Positioning Model

In order to evaluate the average positioning error of the proposed DDAE in the specific-location scenario, we compare DDAE with traditional machine learning algorithms and current advanced deep learning algorithms. The experimental results show that DDAE has obvious advantages with the smallest average positioning error in the specific-location positioning scenario, whether compared with traditional machine learning algorithms or deep learning algorithms. The specific analysis is as follows.

Machine learning algorithms used for comparison include SVM [27], KNN [28], RF [29], DT [30], and GNB [31]. Table 5 shows the comparison results of DDAE and the traditional machine learning algorithms.

Table 5. Specific-location positioning results.

Specific-Location Positioning Algorithm	**DT**	**GNB**	**SVM**	**RF**	**KNN**	**Our**
RMSE (m)	10.42	10.42	8.65	7.14	6.13	6.01

It can be seen from Table 5 that the average positioning error of machine learning methods in specific-location positioning is 6.13 m at the lowest (KNN) and 10.42 m at the highest (DT and GNB), while the DAE model proposed in this paper reduces to 6.01 m. Similarly, the state-of-the-art deep learning algorithms used for comparison include SAEDNN [34], CNNLoc [35], CCpos [36], CHISEL [37], CHISEL-DA [37], and HADNN [38]. Figure 9 shows the comparison results of DDAE and the advanced deep learning algorithms. It can be seen from Figure 9 that the average positioning error of machine learning methods in specific-location positioning is 6.95 m at the lowest (CHISEL-DA) and 12.40 m at the highest (CCpos), while the DAE model proposed in this paper reduces to 6.01 m.

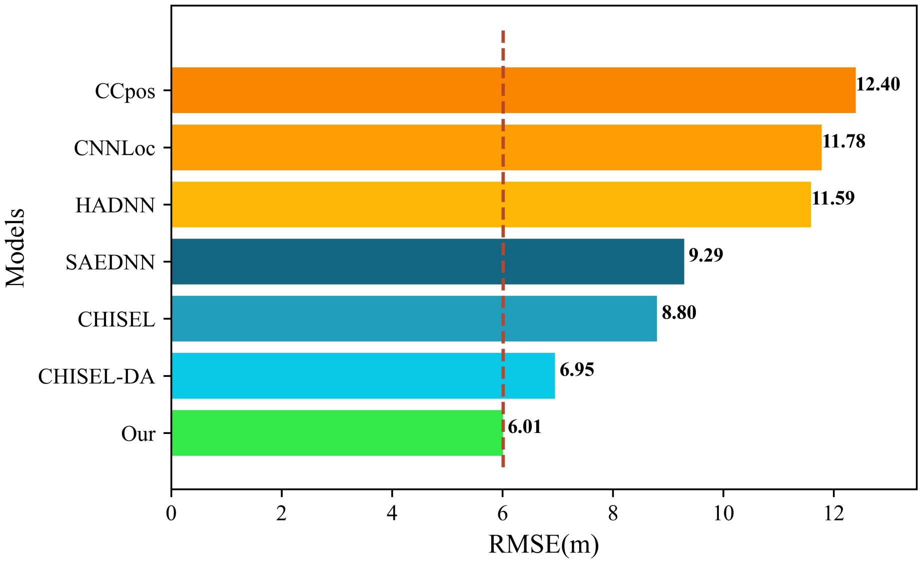


Figure 9. Specific-location positioning error comparison results.

6. Conclusions

In this paper, we propose calibration-free indoor positioning algorithms based on DNN and DIFF, which extend the indoor positioning from a single plane to three dimensions, namely, multi-building, multi-floor, and specific location. However, DIFF brings the explosion of data dimensions, which reduces the positioning efficiency. Therefore, we design a data hierarchical processing strategy based on building-floor-specific location, which effectively improves the efficiency of high-dimensional data processing. Aiming at the issue of positioning accuracy, we design an improved DAE and an enhanced DNN to improve the positioning accuracy of multi-building and multi-floor, respectively. In the specific-location positioning scenario, a newly designed positioning algorithm called DDAE is proposed. The DDAE consists of an optimized DAE and an enhanced DNN. By utilizing the optimized DAE to reduce the dimensions of input data DIFF while retaining the key features and optimizing the structure and parameters of enhanced DNN, the DDAE can obtain better positioning accuracy. Our work addresses both positioning accuracy and positioning efficiency issues. Future works may further consider the adaptability of the proposed algorithms. In addition, the construction of a new fingerprint signal data set to verify the performance of algorithms more conveniently and objectively also needs further consideration.

Author Contributions: Data curation, S.D. and W.Z. ; methodology, S.D. and J.Y. ; Writing—original draft, S.D.; Writing—review & editing, J.Y. and L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fujian Province Nature Science Foundation (No. 2020J01813 and No. 2021J011002), the Zhangzhou Municipal Natural Science Foundation (No. ZZ2021J23), the Opening Foundation of Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund, Fujian Normal University (No. NSCL-KF2021-07), and the Research Project on Education and Teaching Reform of Undergraduate Colleges and Universities in Fujian Province (FBJG20210070).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, F.; Geng, Z. Analysis on the development status and prospect of intelligent medical. In Proceedings of the 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 14–16 April 2021; pp. 416–419. [CrossRef]
2. Maswadi, K.; Ghani, N.B.A.; Hamid, S.B. Systematic literature review of smart home monitoring technologies based on IoT for the elderly. *IEEE Access* **2020**, *8*, 92244–92261. [CrossRef]
3. Chen, C.H.; Lee, C.A.; Lo, C.C. Vehicle localization and velocity estimation based on mobile phone sensing. *IEEE Access* **2016**, *4*, 803–817. [CrossRef]
4. Mohammad, N.; Muhammad, S.; Bashar, A.; Khan, M.A. Formal analysis of human-assisted smart city emergency services. *IEEE Access* **2019**, *7*, 60376–60388. [CrossRef]
5. Chen, H.; Guo, B.; Yu, Z.; Han, Q. Crowdtracking: Real-time vehicle tracking through mobile crowdsensing. *IEEE Internet Things J.* **2019**, *6*, 7570–7583. [CrossRef]
6. Zhu, Q.; Xiong, Q.; Wang, K.; Lu, W.; Liu, T. Accurate WiFi-based indoor localization by using fuzzy classifier and mlps ensemble in complex environment. *J. Frankl. Inst.* **2020**, *357*, 1420–1436. [CrossRef]
7. Li, Z.; Yang, Y.; Pahlavan, K. Using iBeacon for newborns localization in hospitals. In Proceedings of the 2016 10th International Symposium on Medical Information and Communication Technology (ISMICT), Worcester, MA, USA, 21–23 March 2016; pp. 1–5. [CrossRef]
8. Lau, S.L.; Toh, C.; Saleem, Y. Wi-Fi Fingerprint localisation using Density-based Clustering for public spaces: A case study in a shopping mall. In Proceedings of the 2016 6th International Conference—Cloud System and Big Data Engineering (Confluence), Noida, India, 14–15 January 2016; pp. 356–360. [CrossRef]
9. Saab, S.S.; Nakad, Z.S. A Standalone RFID Indoor Positioning System Using Passive Tags. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1961–1970. [CrossRef]
10. Tesoriero, R.; Tebar, R.; Gallud, J.; Lozano, M.; Penichet, V. Improving location awareness in indoor spaces using RFID technology. *Expert Syst. Appl.* **2010**, *37*, 894–898. [CrossRef]
11. Alvarez, Y.; Las Heras, F. ZigBee-based Sensor Network for Indoor Location and Tracking Applications. *IEEE Lat. Am. Trans.* **2016**, *14*, 3208–3214. [CrossRef]
12. Uradzinski, M.; Guo, H.; Liu, X.; Yu, M. Advanced Indoor Positioning Using Zigbee Wireless Technology. *Wirel. Pers. Commun.* **2017**, *97*, 6509–6518. [CrossRef]
13. Jiang, X.; Chen, Y.; Liu, J.; Gu, Y.; Hu, L. FSELM: fusion semi-supervised extreme learning machine for indoor localization with Wi-Fi and Bluetooth fingerprints. *Soft Comput.* **2018**, *22*, 3621–3635. [CrossRef]
14. Khoury, H.M.; Kamat, V.R. Evaluation of position tracking technologies for user localization in indoor construction environments. *Autom. Constr.* **2009**, *18*, 444–457. [CrossRef]
15. Luo, C.; Cheng, L.; Chan, M.C.; Gu, Y.; Li, J.; Ming, Z. Pallas: Self-Bootstrapping Fine-Grained Passive Indoor Localization Using WiFi Monitors. *IEEE Trans. Mob. Comput.* **2017**, *16*, 466–481. [CrossRef]
16. Yang, J.; Liu, W.; Huang, H. *Wireless Network Technology: Principle, Experiment and Network Design*; Xiamen University Press: Xiamen, China, 2018.
17. Sadowski, S.; Spachos, P. RSSI-Based Indoor Localization With the Internet of Things. *IEEE Access* **2018**, *6*, 30149–30161. [CrossRef]
18. Ravi, A.; Misra, A. Practical server-side WiFi-based indoor localization: Addressing cardinality and outlier challenges for improved occupancy estimation. *Ad Hoc Netw.* **2021**, *115*, 102443. doi: 10.1016/j.adhoc.2021.102443. [CrossRef]
19. Gómez-Tornero, J.L.; Cañete-Rebenaque, D.; López-Pastor, J.A.; Martínez-Sala, A.S. Hybrid Analog-Digital Processing System for Amplitude-Monopulse RSSI-Based MiMo WiFi Direction-of-Arrival Estimation. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 529–540. [CrossRef]
20. Guo, X.; Li, L.; Ansari, N.; Liao, B. Accurate WiFi Localization by Fusing a Group of Fingerprints via a Global Fusion Profile. *IEEE Trans. Veh. Technol.* **2018**, *67*, 7314–7325. [CrossRef]
21. Lin, M.; Chen, B.; Zhang, W.; Yang, J. Characteristic analysis of wireless local area network's received signal strength indication in indoor positioning. *IET Commun.* **2020**, *14*, 497–504. [CrossRef]
22. Bae, H.J.; Choi, L. Large-Scale Indoor Positioning using Geomagnetic Field with Deep Neural Networks. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6. [CrossRef]
23. Chow, K.H.; He, S.; Tan, J.; Chan, S.H.G. Efficient Locality Classification for Indoor Fingerprint-Based Systems. *IEEE Trans. Mob. Comput.* **2019**, *18*, 290–304. [CrossRef]
24. Dwiyasa, F.; Lim, M.H. A survey of problems and approaches in wireless-based indoor positioning. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Alcala de Henares, Spain, 4–7 October 2016; pp. 1–7. [CrossRef]
25. Kjærgaard, M.B.; Munk, C.V. Hyperbolic Location Fingerprinting: A Calibration-Free Solution for Handling Differences in Signal Strength (concise contribution). In Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom), Hong Kong, China, 17–21 March 2008; pp. 110–116. [CrossRef]

26. Dong, F.; Chen, Y.; Liu, J.; Ning, Q.; Piao, S. A Calibration-Free Localization Solution for Handling Signal Strength Variance. In *Mobile Entity Localization and Tracking in GPS-Less Environments, Proceedings of the International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments, Orlando, FL, USA, 30 September 2009*; Fuller, R., Koutsoukos, X.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 79–90. [CrossRef]
27. Chriki, A.; Touati, H.; Snoussi, H. SVM-based indoor localization in Wireless Sensor Networks. In Proceedings of the 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain, 26–30 June 2017; pp. 1144–1149. [CrossRef]
28. Torres-Sospedra, J.; Montoliu, R.; Martínez-Usó, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 261–270. [CrossRef]
29. Gorak, R.; Luckner, M. Modified Random Forest Algorithm for Wi–Fi Indoor Localization System. In *Computational Collective Intelligence, Proceedings of the International Conference on Computational Collective Intelligence, Halkidiki, Greece, 28–30 September 2016*; Nguyen, N.T., Iliadis, L., Manolopoulos, Y., Trawinski, B., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 147–157. [CrossRef]
30. Sánchez-Rodríguez, D.; Hernández-Morera, P.; Quinteiro, J.M.; Alonso-González, I. A Low Complexity System Based on Multiple Weighted Decision Trees for Indoor Localization. *Sensors* **2015**, *15*, 14809–14829. [CrossRef]
31. Zhang, G.; Wang, P.; Chen, H.; Zhang, L. Wireless Indoor Localization Using Convolutional Neural Network and Gaussian Process Regression. *Sensors* **2019**, *19*, 2508. [CrossRef]
32. Nowicki, M.; Wietrzykowski, J. Low-Effort Place Recognition with WiFi Fingerprints Using Deep Learning. In *Automation 2017, Proceedings of the International Conference Automation, Warsaw, Poland, 15–17 March 2017*; Szewczyk, R., Zielinski, C., Kaliczynska, M., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 575–584. [CrossRef]
33. Jang, J.W.; Hong, S.N. Indoor Localization with WiFi Fingerprinting Using Convolutional Neural Network. In Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, 3–6 July 2018; pp. 753–758. [CrossRef]
34. Kim, K.S.; Lee, S.; Huang, K. A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on Wi-Fi fingerprinting. *Big Data Anal.* **2018**, *3*, 4. [CrossRef]
35. Song, X.; Fan, X.; He, X.; Xiang, C.; Ye, Q.; Huang, X.; Fang, G.; Chen, L.L.; Qin, J.; Wang, Z. CNNLoc: Deep-Learning Based Indoor Localization with WiFi Fingerprinting. In Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, UK, 19–23 August 2019; pp. 589–595. [CrossRef]
36. Qin, F.; Zuo, T.; Wang, X. Ccpos: Wifi fingerprint indoor positioning system based on cdae-cnn. *Sensors* **2021**, *21*, 1114. [CrossRef] [PubMed]
37. Wang, L.; Tiku, S.; Pasricha, S. CHISEL: Compression-Aware High-Accuracy Embedded Indoor Localization With Deep Learning. *IEEE Embed. Syst. Lett.* **2022**, *14*, 23–26. [CrossRef]
38. Cha, J.; Lim, E. A hierarchical auxiliary deep neural network architecture for large-scale indoor localization based on Wi-Fi fingerprinting. *Appl. Soft Comput.* **2022**, *120*, 108624. [CrossRef]
39. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
40. Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; Manzagol, P.A. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.* **2010**, *11*, 3371–3408. [CrossRef]
41. Moreira, A.; Nicolau, M.J.; Meneses, F.; Costa, A. Wi-Fi fingerprinting in the real world—RTLS@UM at the EvAAL competition. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–10. [CrossRef]
42. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.