applied sciences

MDPI

Article

Robust and Accurate Wi-Fi Fingerprint Location Recognition Method Based on Deep Neural Network

Yifan Wang, Jingxiang Gao *, Zengke Li and Long Zhao

School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou 221116, China; cumtwangyf@163.com (Y.W.); zengkeli@yeah.net (Z.L.); cehuizl@126.com (L.Z.)
* Correspondence: jxgao@cumt.edu.cn; Tel.: +86-516-8388-5785

Received: 4 December 2019; Accepted: 27 December 2019; Published: 1 January 2020

Abstract: Currently, indoor locations based on the received signal strength (*RSS*) of Wi-Fi are attracting more and more attention thanks to the technology's low cost, low power consumption and wide availability in mobile devices. However, the accuracy of Wi-Fi positioning is limited, due to the signal fluctuation and indoor multipath interference. In order to overcome this problem, this paper proposes a robust and accurate Wi-Fi fingerprint location recognition method based on a deep neural network (DNN). A stacked denoising auto-encoder (SDAE) is used to extract robust features from noisy *RSS* to construct a feature-weighted fingerprint database offline. We use the combination of the weights of posteriori probability and geometric relationship of fingerprint points to calculate the coordinates of unknown points online. In addition, we use constrained Kalman filtering and hidden Markov models (HMM) to smooth and optimize positioning results and overcome the influence of gross error on positioning results, combined with characteristics of user movement in buildings, both dynamic and static. The experiment shows that the DNN is feasible for position recognition, and the method proposed in this paper is more accurate and stable than the commonly used Wi-Fi positioning methods in different scenes.

Keywords: indoor location recognition; received signal strength (*RSS*); Wi-Fi fingerprint positioning; deep neural network (DNN); optimization methods; adaptive filter; hidden Markov models (HMM)

1. Introduction

With the popularization of wireless communication equipment and the improvement of location service technology, people are looking for more accurate location recognition technology in more extensive and complex applications [1,2]. Global navigation satellite systems (GNSS) play an important role in outdoor navigation due to their positioning ability, but their use is limited in indoor environments because the signal is blocked by houses and the inhabitants inside. In order to satisfy the needs of indoor location-based services (LBS), such as ultrasound [3], ultra-wideband [4], geomagnetic fields [5], images [6], light [7], inertia systems [8] and wireless signals [9] are used for indoor positioning. As infrastructure for widely deployed public networks, Wi-Fi access point (AP) networking hardware devices are now found in nearly every indoor area where people conduct their daily activities, which provides a huge application space for indoor location services. Therefore, Wi-Fi-based indoor positioning technology has become a preference for different kinds of indoor positioning technologies due to its low cost, low power consumption and wide availability in mobile devices [10].

As with other wireless technologies, time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength (*RSS*) approaches are used to achieve Wi-Fi positioning. Unlike the other three methods of calculating user coordinates through geometric relationships, the Wi-Fi fingerprint location method based on *RSS* uses a matching algorithm to determine location with the character of signal strength. As a result, this method is less affected by the influence of

environmental change, and there is no need to know the AP location, which explains why it has become mainstream Wi-Fi location technology [11]. However, wireless signal strength is not stable in complex indoor environments. Affected by multipath interference, *RSS* may fluctuate greatly, which will reduce its positioning accuracy.

At present, the advancement of micro-electro-mechanical-systems (MEMS) technology has led to more sensors in smartphones [12]. Researchers propose that combining a multi-sensor scheme with Wi-Fi data will profoundly improve the positioning accuracy [13,14]. However, equipment, such as accelerometers, gyroscopes and compasses, is prohibitively expensive for most customers, and the use of multiple sensors will inevitably increase power consumption. At the same time, users only need to know their outline coordinates and cooperate with the visual inspection to determine their location in daily life. Thus, it is more in accordance with users' requirements to obtain rapid and accurate positioning information by relying on fewer sensors in the public environment. Since Wi-Fi positioning has a huge advantage due to the wide deployment of its infrastructure, the focus is on how to improve its positioning performance in buildings. Machine learning has the ability to mine data features, which is a very promising technology for solving the problem of Wi-Fi signal mismatch, and has been the focus of much recent research.

In this paper, a robust and accurate Wi-Fi fingerprint location recognition method based on deep neural network is put forward, and its main contributions are the following:

(1) We use the stacked denoising autoencoders (SDAE) in a deep neural network (DNN) to perform Wi-Fi location recognition. Compared to the traditional Wi-Fi fingerprint location method, the proposed method can fully obtain feature representations from noisy *RSS*, reduce the dimension of the Wi-Fi signal and improve positioning accuracy and robustness. In addition, we use regularization techniques to avoid the model overfitting offline, and take into account the geometric distribution of fingerprint points in order to estimate the user's position and further ensure accuracy during the positioning stage.

(2) We construct a dynamic constrained Kalman filter (KF) based on the speed at which users inside a building move. This ensures the continuity of dynamic positioning results and improves the positioning accuracy by depending on identifying and smoothing out the gross errors caused by multipath interference. The hidden Markov model (HMM) is used to optimize multiple positioning results of the same unknown point to obtain the optimum coordinates of the unknown point based on the temporal correlation of multiple observations that are made when the users are not moving.

(3) We applied the UJIIndoorLoc indoor localization public dataset [15] to fully verify the feasibility of this method in floor recognition and location estimation, and analyzed the results at length. Moreover, we established an experimental site in an office building to test the positioning performance of constrained KF and HMM under dynamic and static conditions, respectively.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 provides the relevant methodology and more details about DNN-based Wi-Fi location recognition, constrained KF and HMM. Section 4 discusses the experimental results of the proposed method with different modes. Section 5 contains our conclusions, as well as suggestions for future research.

2. Related Works

In 2000, Microsoft developed RADAR, a radio frequency (RF)-based wireless local area network (WLAN) indoor positioning system for locating and tracking users inside buildings [16]. It used a k-nearest neighbors (KNN) approach, which used the European distance of signal space to match and locate, and became the gold standard of fingerprint positioning. Clustering-KNN [17] and improving weighted KNN [18] approached soon followed, but more time was needed to deal with the challenge of row data with the aim of improving accuracy. The traditional methods were unable to accomplish this because of the complex nature of the signal error mechanism [19]. In [20], the authors constructed

a fingerprint system with the help of Wi-Fi channel State information (CSI), based on a neural network. The system achieved better accuracy, but because special equipment was required to obtain CSI, it was not considered a universal solution.

Wi-Fi signals are increasingly being received in public places. However, it is a challenge to use this big data fully and effectively, although the rapid development of machine learning theory makes it possible because it requires less parameter tuning and offers better scalability. Support vector machine [21] and random forest [22] have been used to capture the correlation between signal strength and physical location. Michael et al. [23] first used a DNN to realize floor classification based on Wi-Fi, but the solution could not estimate user location. In [24], the authors realized Wi-Fi fingerprint location based on deep learning by simulation, with location error less than 1 m. Zhang et al. [25] and Rizk et al. [26] proposed DNN-based solutions that extracted features from large noise signal samples for localization. They analyzed the impact of model parameters and data structures on positioning performance. With the gradual combination of crowdsourcing theory with wireless location, it is necessary to explore techniques for deep learning-based indoor positioning under the background of big data to obtain the better positioning accuracy and data utilization.

Multipath and non-line-of-sight (NLOS) propagation inevitably leads to gross errors in the Wi-Fi positioning results in complex indoor environments. However, many constraints exist when it comes to obtaining information about user navigation, and some of them can be used to improve the positioning accuracy. A smoothed constrained fingerprinting solution has also been adopted in a system that integrates a dual-filter MEMS sensor and Wi-Fi fingerprinting [27]. Poulose et al. [28] combined the Pedestrian Dead Reckoning localization algorithm with the linear Kalman filter to achieve Wi-Fi fusion positioning results. Li et al. [29] used fingerprinting accuracy indicators (FAIs) to predict wireless and magnetic fingerprinting accuracy, and then used the FAI-enhanced EKF to improve the stability of multi-sensor fusion. In [30], Bi et al. proposed an adaptive weighted KNN Wi-Fi location method, which improved the accuracy of single point positioning with the constraints of the user's orientation and the hybrid distance of the signal domain and position domain. In [26], the authors set up the CellinDeep system-based Wi-Fi signal, which used data augmenter techniques and mean filtering to address practical challenges caused by signal noise. Zhou et al. [31] used Kalman filtering to reduce drift and oscillation on a received signal strength indicator (RSSI) caused by the noise overlap-add in Bluetooth locations. Nevertheless, most of these methods rely on multiple sensors, and there has been relatively little research on a single Wi-Fi constraint location. Those Wi-Fi experiments that have taken place are based on discrete points. Inspired by the above research, we propose a robust and accurate Wi-Fi fingerprint location recognition method based on a DNN, along with optimization methods, which are adaptive constrained KF and HMM, to improve the accuracy and stability of a single Wi-Fi location on the basis of different movement modes of users.

3. Theoretical Frameworks

3.1. Overview

Figure 1 shows the architecture of Wi-Fi fingerprint location recognition method based on a DNN. The method has two phases: an offline training phase and online positioning phase. During the offline phase, the raw *RSS* of different Wi-Fi APs on the selected fingerprint points in the area of interest is recorded. We then trained the DNN and extracted features from the noisy *RSS* to construct an offline feature-weighted fingerprint database. During the online phase, the real-time AP information observed on unknown points is transferred to the feature fingerprint database trained in the offline phase to estimate the floor or location of the current user. We chose constrained Kalman filtering and hidden Markov model (HMM) to optimize the results of dynamic and static positioning, respectively, in order to reduce the number of outliers and obtain more accurate positioning results with regard to different motion modes of users.

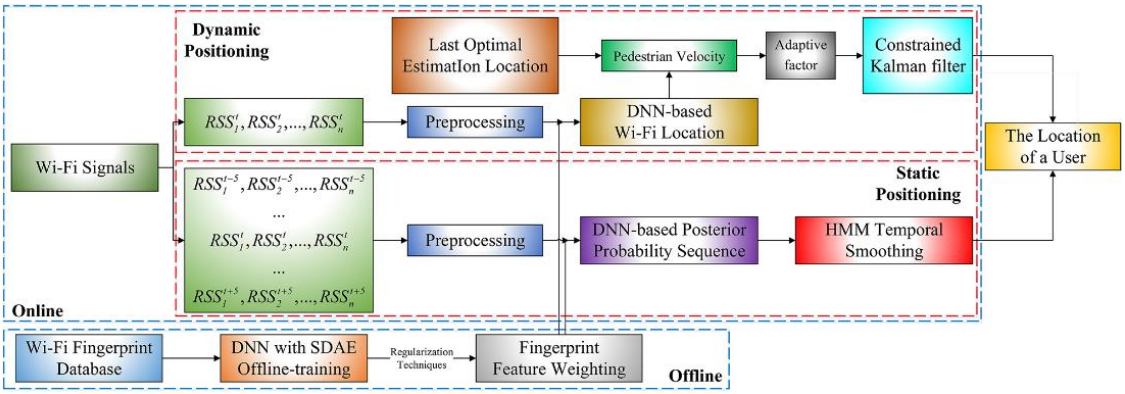

Figure 1. System framework.

3.2. DNN-Based Wi-Fi Location Recognition

3.2.1. Data Preprocessing

When every AP is observed, a pair information <MAC *RSS*> is recorded, in which MAC represents its unique ID and *RSS* its received signal strength. All AP information that can be observed in each scan is recorded as Wi-Fi training fingerprint or test data.

The preprocessing mechanism runs during both the offline and online stages. The first 15 APs with the largest *RSS* in every scan are retained to make the *RSS* set input to the depth model consistent, $o = \{rss_1, rss_2, \ldots, rss_{15}\}$, where each entry represents observation information in the test area. But it should be noted that there are only 15 currently observed elements in each recorded sample. Therefore, for the entire training set, all the elements in the sample are arranged uniformly in the AP list in the fingerprint database, and the *RSS* of the unobserved AP in each input is set at −100 dB.

We standardized the observed raw *RSS*, which ranges from −100 dB to 0 dB, to adjust its value to between 0 and 1 in order to reduce the distribution of *RSS* and facilitate the subsequent training. Formula (1) shows the standardization process. Formula (2) shows the fingerprint database built after standardizing the *RSS* of all fingerprint points.

$$RSS_i' = \frac{RSS_i + 100}{\max(RSS) + 100} \tag{1}$$

$$\textit{Fingerprint Database} \rightarrow \begin{bmatrix} RSS_1^{1\prime} & RSS_1^{2\prime} & RSS_1^{3\prime} & \cdots & RSS_1^{n\prime} \\ RSS_2^{1\prime} & RSS_2^{2\prime} & RSS_2^{3\prime} & \cdots & RSS_2^{n\prime} \\ \cdots & \cdots & \cdots & \ddots & \cdots \\ RSS_{z-1}^{1\prime} & RSS_{z-1}^{2\prime} & RSS_{z-1}^{3\prime} & \cdots & RSS_{z-1}^{n\prime} \\ RSS_z^{1\prime} & RSS_z^{2\prime} & RSS_z^{3\prime} & \cdots & RSS_z^{n\prime} \end{bmatrix} \tag{2}$$

where RSS_i' represents the standardized value, $\max(RSS)$ is the maximum in the fingerprint database, n the number of APs observed in offline training, z the number of fingerprint points. Every row is an observation sample, and the *RSS* of the unobserved AP is set to −100 dB.

3.2.2. Offline Stage

The SDAE is used to capture nonlinear and implicit correlations between *RSS* and the spatial location through signal changes at different fingerprint points. The SDAE is able to represent input data in the form of hierarchically distributed data, using the observation information on fingerprint points and reducing the feature dimension of signal data [32].

An SDAE is stacked by multiple denoising autoencoders (DAEs) as hidden layers. Each DAE layer is used to train the pair of encoder-decoders to make the output value u equal to the input value

x as much as possible. If input data can be restored after encoding and decoding, then the feature representations trained in this way will contain relatively low noise levels. Figure 2 shows the process of a DAE. The encoder-decoder can be formulated as

$$\begin{cases} y(x) = \frac{1}{1+e^{-(wx+b)}} \\ u(y) = \frac{1}{1+e^{-(w'y+b')}} \end{cases} \tag{3}$$

where w and w' are the connection weights between the encoder and decoder, respectively; they are the transposition of each other, and b denotes the hidden units biases, while b' denotes the visible units biases.

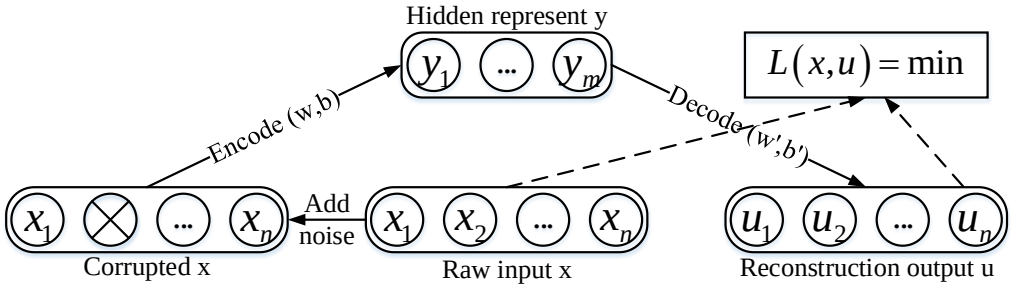

Figure 2. Training process of a denoising autoencoder (DAE).

We employed two techniques to suppress overfitting in training the DNN. First, we added a noise reduction mechanism during training [33]. Before the *RSS* in the fingerprint database is input into each DAE for unsupervised training, it will be "corrupted" by random setting to 0 based on a certain probability. The trained features are of higher robustness since corrupted input data are closer to actual test data. Second, we adopted dropout regularization, which has been shown to be useful for large networks [34]. This method randomly removed some neurons in different parts of the training stage, which means these neurons cannot participate in propagation. This prevented the interdependence of neurons because the DNN relied too much on several local features in the training process, and also enhanced the generalization of the model.

The softmax function is usually used for solving the finite discrete probability distribution in classification. After layer-by-layer unsupervised training of the DNN's three hidden layers, we treated the softmax function as the final output layer and prior probability of all fingerprint points as labeled information in consideration of classification. The feature weight obtained from the training was recorded to complete the offline training, but not before being fine-tuned from top to bottom.

The method constructs a four-layer DNN used for location recognition, as is shown in Figure 3. Boxes 1, 2 and 3 represent the hidden layers of DAE respectively, and the last box 4 represents the softmax output layer. After the standardized input *RSS X'* enters DNN, the training of each DAE hidden layer is completed according to the procedures shown in Figure 2 in turn, and the trained connection weight w and hidden units biases b are saved. The output y of the previous layer is taken as the input of the next layer to propagate layer by layer. When data reaches the last layer, the priority probability of all fingerprint points as labeled is used for global fine-tuning from top to bottom to realize offline training. N is the neuron dimension of input layer, and the number is equal to APs observed in offline training. Z is the output layer dimension and is equal to fingerprints when the model is used for positioning, and is equal to floors when the model is used for floor recognition. H_1, H_2 and H_3 represent the neuron dimension of each hidden layer. Their value is determined by the actual situation.

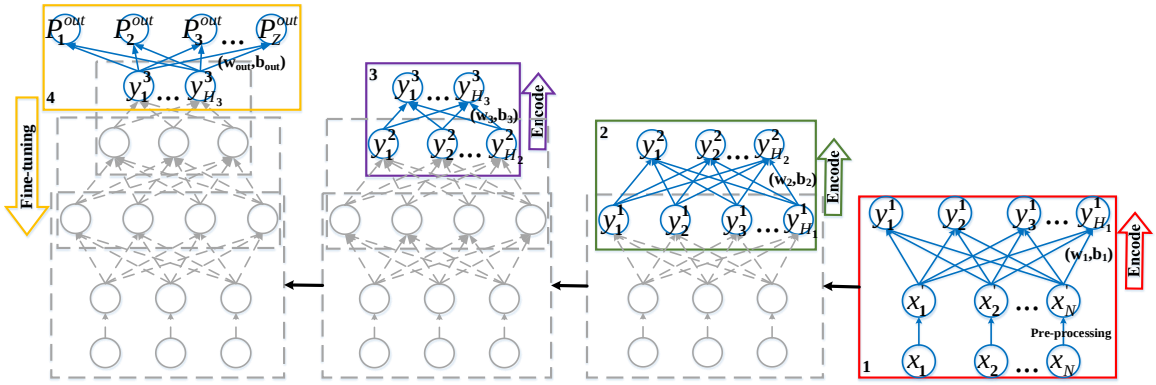

Figure 3. Deep neural network (DNN) structure used for location recognition.

3.2.3. Online Stage

In online positioning, we only need to propagate the preprocessed real-time observation $x^L = (RSS'_1, RSS'_2, \ldots, RSS'_n)$ from bottom to top into the trained DNN in the offline phase, without additional screening and clustering. The propagation process is shown as Formula (4), and y^1, y^2 and y^3 are the output of each hidden layer.

$$\begin{cases} y^1\left(x^L\right) = \frac{1}{1+e^{-(w_1 x^L + b_1)}} \\ y^2\left(y^1\right) = \frac{1}{1+e^{-(w_2 y^1 + b_2)}} \\ y^3\left(y^2\right) = \frac{1}{1+e^{-(w_3 y^2 + b_3)}} \end{cases} \tag{4}$$

The output feature y^3 is then passed into softmax output layer. According to Formula (5), we calculate the posterior probability $P(l_i|L, out), i = 1, 2, \ldots, z$ of the unknown point L being at each fingerprint point l_i:

$$P(l_i|L, out) = \frac{e^{\left(-w_i^{out} y^3 - b_i^{out}\right)}}{\sum_{j=1}^{z} e^{\left(-w_j^{out} y^3 - b_j^{out}\right)}}, i = 1, 2, \ldots, z \tag{5}$$

During the process of floor recognition, the user's floor is the same as the floor with the maximum probability. During the estimate of positioning, the top K fingerprint points of maximum probability are selected as neighboring points to calculate the coordinates of the unknown point. In theory, neighboring points should be distributed around the unknown point. Taking the geometric relationship into consideration, we calculate the center of neighboring points and take the geometric distance between the center and each neighboring point as the weight factor. The larger the distance, the less significant the relationship between the neighboring point and its center, and the smaller the effect of the neighboring point on the unknown point. Neighboring points that are significantly far away from the center are removed. The coordinates of the unknown point are then calculated by combination weights of geometric distance and posterior probability of selected neighboring points. Figure 4 shows the process of combination weight determination, which is based on the following:

$$\left(X_{center}, Y_{center}\right) = \frac{\sum_{i=1}^{K} \left(X_{l_i}, Y_{l_i}\right)}{K} \tag{6}$$

$$Distance_i = \left(\left(X_{l_i} - X_{center}\right)^2 + \left(Y_{l_i} - Y_{center}\right)^2\right)^{\frac{1}{2}}, i = 1, 2, \cdots, K \tag{7}$$

$$W_i = \frac{1/Distance_i + P(l_i|L, out)}{\sum_{j=1}^{j=K}\left(1/Distance_j + P\left(l_j|L, out\right)\right)}, i = 1, 2, \ldots, K \tag{8}$$

$$(X_L, Y_L) = \sum_{i=1}^{K}\left(W_i\left(X_{l_i}, Y_{l_i}\right)\right) \tag{9}$$

where (X_{center}, Y_{center}) is the center coordinates of the neighboring points; W_i is the combination weight of the neighboring point l_i with respect to the unknown point L.

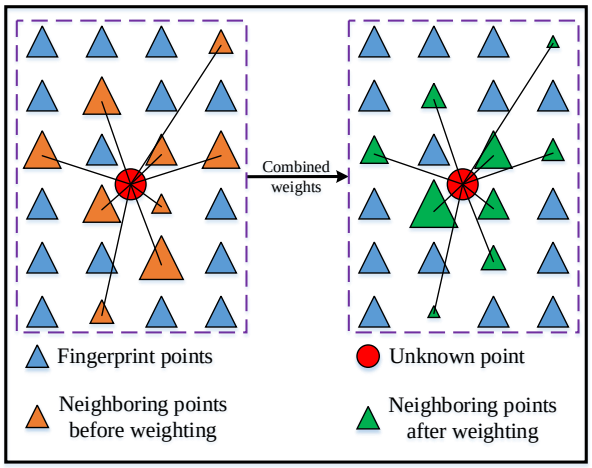

Figure 4. Process of the combination weighting method.

3.3. The Fine Localizer

The motion state is mainly divided into dynamic and static for the user in an indoor environment. According to the dynamic and static characteristics, we propose constrained KF and HMM, respectively, to further improve the accuracy of DNN-based Wi-Fi location estimation, under the condition of only using Wi-Fi.

3.3.1. Constrained Kalman Filter Smoothing

People usually move slowly in an indoor environment. We used KF to solve the smoothing problem in dynamic Wi-Fi location estimation to obtain the optimal estimation of a linear system. We took minimized mean square error as the criterion, estimated the current state by the previous optimal position, used the DNN-based Wi-Fi positioning results as the observation, then updated the information gain to obtain the optimal estimate of the current location.

In this paper, the state vector consists of user positions, velocities and accelerations, and is represented as

$$\boldsymbol{X} = [N, E, V_N, V_E, a_N, a_E] \tag{10}$$

In addition, the state estimation equation of the user at each time is expressed as:

$$\boldsymbol{X}_{t+1,t} = \boldsymbol{A}_{t+1}\boldsymbol{X}_{t,t} + \boldsymbol{Q} \Rightarrow \begin{cases} N_{t+1,t} = \hat{N}_{t,t} + \hat{V}^N_{t,t} * \triangle t + dw_N \\ E_{t+1,t} = \hat{E}_{t,t} + \hat{V}^E_{t,t} * \triangle t + dw_E \\ V^N_{t+1,t} = \hat{V}^N_{t,t} + \hat{a}^N_{t,t} * \triangle t + dw^N_V \\ V^E_{t+1,t} = \hat{V}^E_{t,t} + \hat{a}^E_{t,t} * \triangle t + dw^E_V \\ a^N_{t+1,t} = \hat{a}^N_{t,t} + dw^N_a \\ a^E_{t+1,t} = \hat{a}^E_{t,t} + dw^E_a \end{cases} \tag{11}$$

where dw_N, dw_E, dw_V^N, dw_V^E, dw_a^N and dw_a^E denote the process noises, have a Gaussian distribution. The standard deviations matrix of the process noises is written as $\boldsymbol{Q} = diag[\delta w_N{}^2 \delta w_E{}^2\ \delta w_V^{N2}\ \delta w_V^{E2}\ \delta w_a^{N2}\ \delta w_a^{E2}]$, given $\delta w_N = \delta w_E = 3$, $\delta w_V^N = \delta w_V^E = 0.5$, $\delta w_a^N = \delta w_a^E = 0.1$.

The observation information consists of DNN-based Wi-Fi positioning results and the current user velocities. Therefore, the measurement equation of the system is:

$$\boldsymbol{Z}_{t+1} = \begin{bmatrix} N_{t+1}^{Wi-Fi} \\ E_{t+1}^{Wi-Fi} \\ V_{t+1}^{N} \\ V_{t+1}^{E} \end{bmatrix} = \boldsymbol{H}\boldsymbol{X}_{t+1,t} + \boldsymbol{R} \tag{12}$$

where $\boldsymbol{H} = [diag[1\ 1\ 1\ 1]0_{4\times2}]_{4\times6}$, and $\boldsymbol{R} = diag[\delta N^2\ \delta E^2\ \delta V_N^2\ \delta V_E^2]$ is measurement noise matrix, which has a Gaussian distribution, given $\delta N = \delta E = 12$, $\delta V_N = \delta V_E = 3$. The real-time velocities V_{t+1}^N and V_{t+1}^E are calculated by the current Wi-Fi positioning results and the last optimal location. The calculation formula is as follows:

$$\begin{cases} V_{t+1}^N = (N_{t+1}^{Wi-Fi} - \hat{N}_{t,t}) / \triangle t \\ V_{t+1}^E = (E_{t+1}^{Wi-Fi} - \hat{E}_{t,t}) / \triangle t \end{cases} \tag{13}$$

In the dynamic Wi-Fi positioning smoothing system, the Kalman filter is always adopted to update the state parameters by time and observation updates when Wi-Fi signals are available. In the prediction stage, we predict the state information and calculate its error covariance. The time-update method is expressed as:

$$\bar{\boldsymbol{X}}_{t+1,t} = \boldsymbol{A}_{t+1}\hat{\boldsymbol{X}}_{t,t} \tag{14}$$

$$\bar{\boldsymbol{P}}_{t+1,t} = \boldsymbol{A}_{t+1}\boldsymbol{P}_t\boldsymbol{A}_{t+1}^{\mathrm{T}} + \boldsymbol{Q} \tag{15}$$

We update the measurement to calculate KF gain in the correction stage. The estimation and the error covariance are updated according to real-time observation information. The Kalman filter observation-update equation is:

$$\bar{\boldsymbol{V}}_{t+1} = \boldsymbol{Z}_{t+1} - \boldsymbol{H}\bar{\boldsymbol{X}}_{t+1,t} \tag{16}$$

$$\boldsymbol{P}_{\bar{\boldsymbol{V}}_{t+1}} = \boldsymbol{H}\bar{\boldsymbol{P}}_{t+1,t}\boldsymbol{H}^{\mathrm{T}} + \boldsymbol{R} \tag{17}$$

$$\boldsymbol{G}_{t+1} = \bar{\boldsymbol{P}}_{t+1,t}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}^{-1}_{\bar{\boldsymbol{V}}_{t+1}} \tag{18}$$

$$\hat{\boldsymbol{X}}_{t+1,t+1} = \bar{\boldsymbol{X}}_{t+1,t} + \boldsymbol{G}_{t+1}\bar{\boldsymbol{V}}_{t+1} \tag{19}$$

$$\boldsymbol{P}_{t+1} = (\boldsymbol{I} - \boldsymbol{G}_{t+1}\boldsymbol{H})\bar{\boldsymbol{P}}_{t+1,t} \tag{20}$$

where $\boldsymbol{P}$ represent the covariance matrices of the state vector; $\boldsymbol{G}_{t+1}$ is the gain matrix of the Kalman filter; and subscript $t+1, t$ represents the state or covariance estimates forward from $t+1$ to t. Therefore, we can obtain the optimal location according the optimal estimation $\hat{\boldsymbol{X}}_{t+1,t+1}$ of the system at time $t+1$.

Wi-Fi positioning belongs to absolute positioning, which has a significant impact on the final smoothing results. Therefore, the initial measurement noise $\boldsymbol{R}$ given in the KF smoothing system is relatively small. However, the interfered Wi-Fi signal will inevitably bring gross error to the positioning results in complex indoor environments. Further, the fixed measurement noise will affect the filter smoothing result. Accordingly, we add adaptive constraint on the basis of KF to automatically adjust the measurement noise in the system when detecting the abnormal observation information and thereby improve the positioning accuracy.

We regard the movement of low-speed users in a short time period as a uniform motion, and the velocity information as a constraint condition. When the gross error occurs, the two adjacent positioning results with low-speed motion will be far away from each other, which leads to an abnormal increase in the real-time velocities. We calculate the average of the past three velocities and use that as the standard to judge whether the current velocities are abnormal.

$$\bar{|\boldsymbol{V}_t|} = \frac{1}{3}\sum_{i=t-3}^{t-1}|\boldsymbol{V}_i| \tag{21}$$

When the real-time velocity is less than the standard $\bar{|\boldsymbol{V}_t|}$, it is considered that the observation information is normal, and the measurement noise remains unchanged. Otherwise, it is considered that the Wi-Fi positioning results in gross error and measurement noise need to be enlarged to make the final positioning result of the system closer to the state estimation and reduce the gross error interference. We use the three-segment method to realize the adaptive calculation of constraint factor μ; setting $k_1 = 3\bar{|\boldsymbol{V}_t|}, k_0 = \bar{|\boldsymbol{V}_t|}$, the formula is:

$$\mu = \begin{cases} 1, & |\boldsymbol{V}_t| < k_0 \\ \frac{k_1 - k_0}{k_1 - |\boldsymbol{V}_t|}, & k_0 < |\boldsymbol{V}_t| < k_1 \\ 2, & |\boldsymbol{V}_t| > k_1 \end{cases} \tag{22}$$

If the measurement noise before the constraint is $\boldsymbol{R} = diag[\delta N^2\, \delta E^2\, \delta V_N^2\, \delta V_E^2]$, then the measurement noise is $\boldsymbol{R} = diag[\mu_N * \delta N^2 \mu_E * \delta E^2 \mu_N * \delta {V_N}^2 \mu_E * \delta {V_E}^2]$ after the constraint factor is added.

3.3.2. Hidden Markov Models (HMM) Optimization

The location of users in an indoor environment is uncertain; they may appear anywhere. At this point, users no longer need precise motion trajectory, but they do need accurate coordinates of their current position. However, as a form of absolute positioning, Wi-Fi positioning system cannot converge to a better accuracy as GPS does with the prolongation of observation time. Therefore, we construct HMM for optimization according to the temporal correlation of multiple observations at the same unknown point to improve the accuracy of static positioning and make effective use of the observation rather than simply calculating their mean location.

HMM is a probability model related to time series. We use the current *RSS*, o_t, with its five prior *RSS*, $o_{t-5}, \ldots o_{t-1}$, and five posterior *RSS*, $o_{t+1}, \ldots, o_{t+5}$, as observations at an unknown point. We take fingerprint points as hidden state and the posterior probability calculated by DNN as observation state to structure HMM $\lambda = (A, B, \pi)$.

- $A = (a_{ij}), i, j = 1, 2, \ldots, z$ is the hidden state transition matrix, which is made up of the transition probability among hidden states, and a_{ij} represents the probability of fingerprint point i transferring to j. We set the a_{ij} of non-adjacent fingerprint points to 0 and to 1 otherwise.
- $B = (b_{j\gamma}), j = 1, 2, \ldots z, \gamma = t-5, \ldots, t+5$ is the observation probability matrix, where $b_{j\gamma}$ denotes the probability of o_γ observed at the fingerprint point j. So, $b_{jr} = P(\gamma|l_j) = cP(l_j|\gamma, out)$, where $P(l_j|\gamma, out)$ is the posteriori probability of DNN output, and $c = P(\gamma)/P(l_j)$ is a constant that has nothing to do with final result.
- π is the prior state probability matrix. It represents the probability of the unknown point in each fingerprint point at the start time. In the system, we set the initial probability of each fingerprint point as equal, so $\pi = [1/z, \ldots, 1/z]$.

Therefore, the static optimal posterior probability $P(l_i|L, optimal\ out)$ of the unknown point L at each fingerprint point can be calculated by the forward and backward probability of the HMM when given the *RSS* observations $O = (o_{t-5}, \ldots o_{t-1}, o_t, o_{t+1}, \ldots, o_{t+5})$ and HMM model λ, as follows:

$$P(l_i|L, optimal\ out) = P_t(L_t = l_i, \lambda, O) = \frac{P(L_t = l_i, O|\lambda)}{P(O|\lambda)} = \frac{\mu_t(l_i)\rho_t(l_i)}{\sum_{j=1}^{z} \mu_t(l_j)\rho_t(l_j)}, i = 1, 2, 3, \ldots, z \quad (23)$$

$$\mu_\gamma(l_i) = P(O_{t-5}, \ldots, O_t, L_t = l_i|\lambda) = [\sum_{j=1}^{z} \mu_{\gamma-1}(j)a_{ji}]b_{i\gamma}, \gamma = t-5, \ldots, t-1, t \quad (24)$$

$$\rho_\gamma(l_i) = P(O_t, \ldots, O_{t+5}, L_t = l_i|\lambda) = \sum_{j=1}^{z} a_{ij}b_{j\gamma+1}\rho_{\gamma+1}(j), \gamma = t+5, \ldots, t+1, t \quad (25)$$

where $\mu_t(l_i)$ represents the forward probability of unknown point L at fingerprint point l_i with the observation sequence $o_{t-5}, \ldots o_{t-1}, o_t$, while $\rho_t(l_i)$ is the backward probability of the observation sequence $o_t, o_{t+1}, \ldots, o_{t+5}$ given unknown point L at fingerprint point l_i. Substituting Equation (23) into (8), we can calculate the optimal static coordinates of the unknown point.

4. Experience and Analysis

In order to evaluate the performance of the proposed DNN-based Wi-Fi fingerprint location recognition method, we carried out practical experiments with different scenarios in typical indoor environments. We first used the UJIIndoorLoc public dataset to prove the feasibility of the proposed method when compared against typical Wi-Fi positioning methods, and analyzed the effect of different model parameters on its performance. Next, we established the experiment site in order to perform further testing under the condition of avoiding the influence of heterogeneous equipment. We investigated user positioning performance in dynamic and static states and tested the effectiveness of optimization method on positioning results.

4.1. Experiments of UJIIndoorLoc Public Dataset

4.1.1. Experimental Setting

To evaluate the feasibility of the proposed method, we used the large Wi-Fi crowd-sourced UJIIndoorLoc dataset, which contains labeled positions and is publicly available. The dataset consists of 21,048 Wi-Fi *RSS* samples, of which 19,937 are training samples and 1111 are test samples. Each sample contains 520 *RSS* values from 520 APs. All the registered data were collected by 20 volunteers using 25 different Android devices on 13 floors of three buildings. (The reader is referred to Reference [15] for further details.)

Therefore, the established DNN contained 520 neurons in its input layer. We take every sample as a fingerprint to build the fingerprint database, and the *RSS* of the undetected AP that has positive value 100 in the sample is set to −100 dB. In an SDAE, we set the corrupted probability to 10%; the learning rate and scaling factor to 0.1 and 0.99, respectively; the non-sparse penalty to 0.05; and the dropout fraction to 0.2.

4.1.2. Results of Floor Recognition

Figure 5 shows the recognition results of several DNN architectures with different hidden layers and neurons. As the numbers of hidden layers and neurons increase, the recognition effect improves. However, if the network depth is increased optionally, not only is the running time increased, but the accuracy of recognition is also affected due to limited samples in the fingerprint database. The proposed four-layer DNN architectures obtained the highest recognition rate, as high as 94.60%. We next compared DNN with the standard neural network (NN), which was used for classification. Because the NN lacks the ability to effectively capture feature information, its recognition results were worse than those of the DNN under the same conditions. Compared with standard NN, the recognition rates

of DNN under the one, two, three and four hidden layers architectures were increased 2.11%, 7.91%, 9.71% and 8.12%, respectively. In floor recognition, DNN obtains satisfactory recognition accuracy.

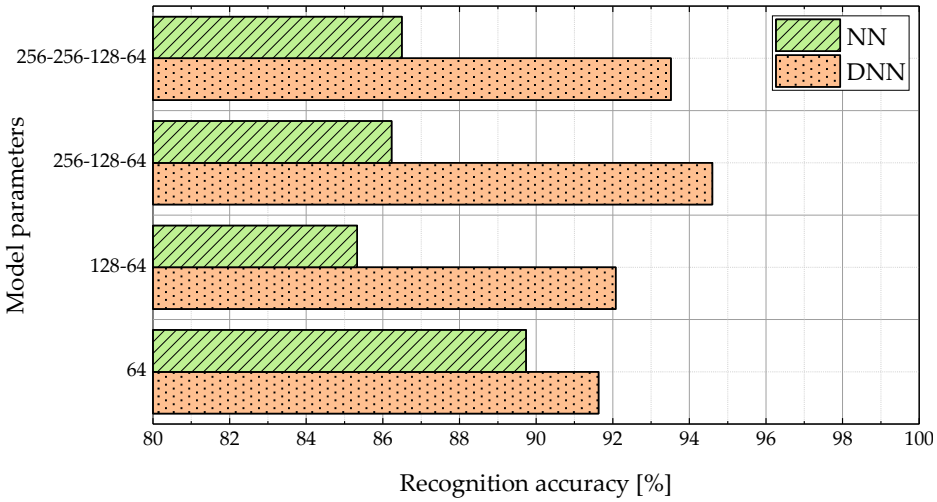

Figure 5. Recognition results for different architectures.

Figure 6 shows the effect of the dropout mechanism on recognition performance in the three-hidden-layer network. Dropout suppresses some neurons in the training process, reduces the model reliance on local features and enhances the generalization of the model. The figure shows that the recognition performance is highest when the dropout mechanism is at 0.2, and that excessive dropout reduces the accuracy of the model. This can be explained by noting that in a large sample model, dropout has a significant effect on restraining overfitting, while excessive dropout destroys the integrity of the model and makes it underfit with the training data.

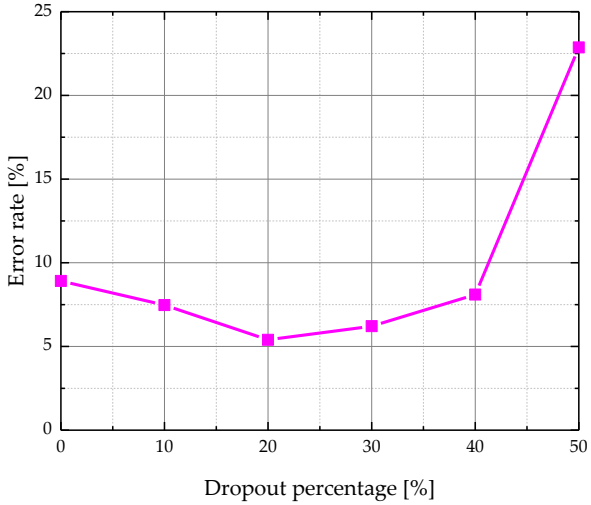

Figure 6. Effect of the dropout on recognition accuracy.

4.1.3. Results of Position Estimation

We chose all floors data to fully test the feasibility of DNN in position estimation and make it more effective. Tables 1–3 show the average error, root-mean-square error (RMSE), maximum error and minimum error of the DNN, as well as those of Microsoft's RADAR system. Figure 7 shows the positioning results of DNN-based Wi-Fi fingerprint location. The positioning error of RADAR is similar to that of Reference [15], while the DNN-based Wi-Fi fingerprint positioning results are better than those of the RADAR system. Compared to the RADAR system, the average errors of DNN

on each floor were reduced as follows: (a) 7.90%, 5.20%, 9.13% and 3.50% for building 0 (Table 1); (b) 56.43%, 21.03%, 5.07% and 15.00% for building 1 (Table 2); (c) 3.66%, 15.98%, 8.64%, 27.03% and 1.32% for building 2 (Table 3). In general, the DNN-based method estimates the location of users better than the RADAR system. The localization results for different floors and different buildings show that the proposed location recognition method is highly robust in different scenes.

Table 1. Positioning errors of Building 0, m.

Floors	Methods	Average Error	Maximum Error	Minimum Error
Floor_0	RADAR	7.72	42.65	0.60
	DNN	7.11	42.79	0.22
Floor_1	RADAR	5.58	33.91	0.11
	DNN	5.29	29.87	0.14
Floor_2	RADAR	6.03	26.49	0.13
	DNN	5.48	23.88	0.33
Floor_3	RADAR	7.14	31.23	0.08
	DNN	6.89	32.75	0.22

Table 2. Positioning errors of Building 1, m.

Floors	Methods	Average Error	Maximum Error	Minimum Error
Floor_0	RADAR	19.14	75.99	0.46
	DNN	8.34	25.98	0.85
Floor_1	RADAR	13.17	54.90	0.19
	DNN	10.40	62.63	0.31
Floor_2	RADAR	13.21	54.91	0.26
	DNN	12.54	55.53	0.92
Floor_3	RADAR	10.27	58.12	0.37
	DNN	8.73	58.39	0.23

Table 3. Positioning errors of Building 2, m.

Floors	Methods	Average Error	Maximum Error	Minimum Error
Floor_0	RADAR	12.57	77.16	0.94
	DNN	12.11	74.53	0.41
Floor_1	RADAR	9.20	68.21	0.40
	DNN	7.73	55.06	0.47
Floor_2	RADAR	12.97	65.71	0.48
	DNN	11.85	52.12	0.79
Floor_3	RADAR	11.06	34.42	9.31
	DNN	8.07	33.83	0.41
Floor_4	RADAR	15.88	74.96	1.39
	DNN	15.67	59.99	0.05

By comparing the positioning results of the three buildings, we found that the positioning accuracy differed for each building, but the positioning results for each floor in the same building were similar. These findings mainly resulted from the deviation between the training data and the test data caused by different acquisition devices [35]. This deviation is a hot spot in the field of research on Wi-Fi signals. Many researchers have attempted to study its mechanism and solution. The relationship between the number of data acquisition devices and positioning errors is shown in Figure 8. It is apparent that the fewer devices used during the acquisition of training data, the higher the positioning accuracy. But this is not the focus of this paper, and therefore we do not carry out a detailed study. In addition, the test data were obtained four months after the original data gathering, which also contributed to increase of error. Therefore, although using the data for positioning is a challenge, DNN-based Wi-Fi

fingerprint location recognition is more accurate than that of the widely used RADAR system, which proves the feasibility and robustness of this method.

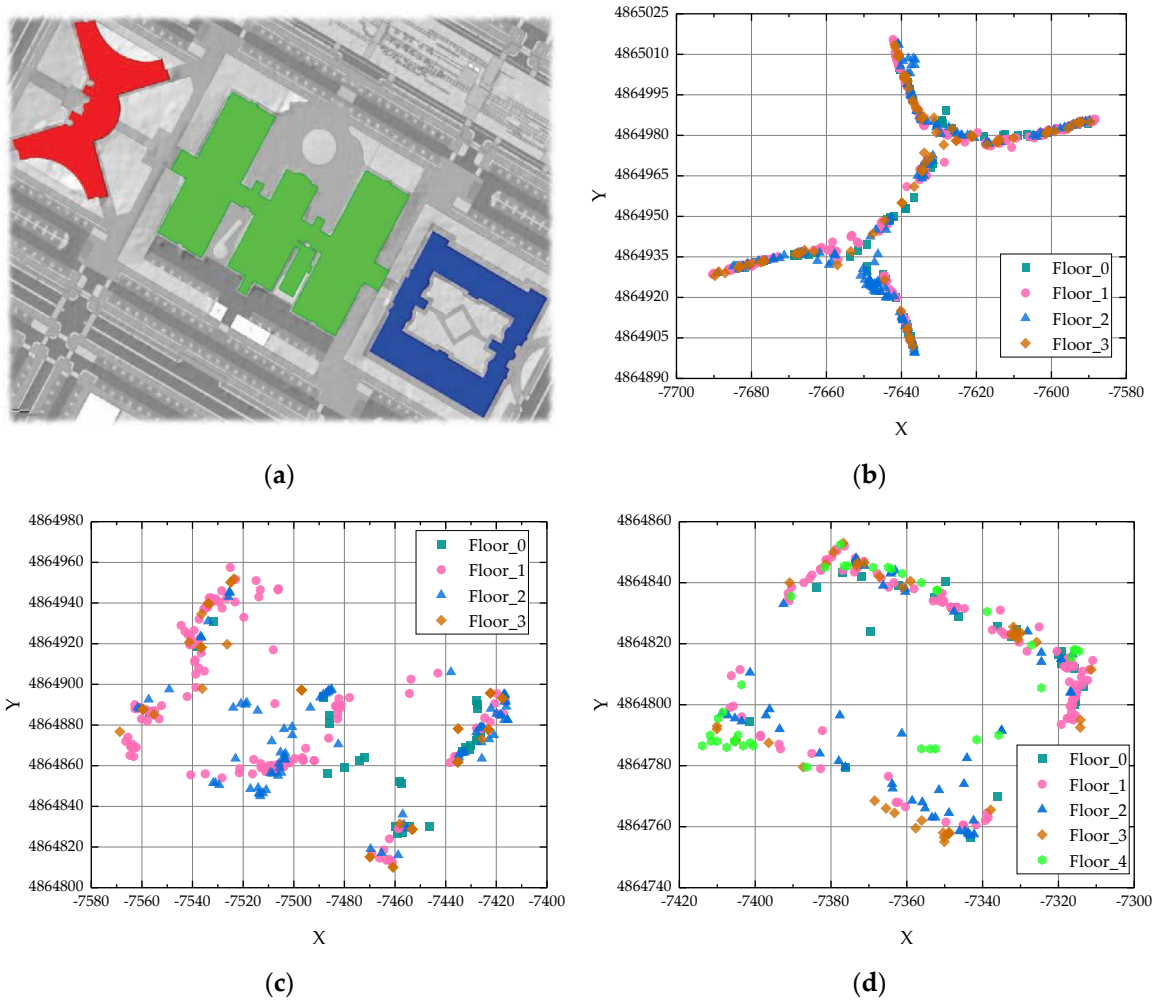

Figure 7. Positioning results of UJIIndoorLoc Public Dataset: (**a**) Building contours; (**b**) Building 0; (**c**) Building 1; (**d**) Building 2.

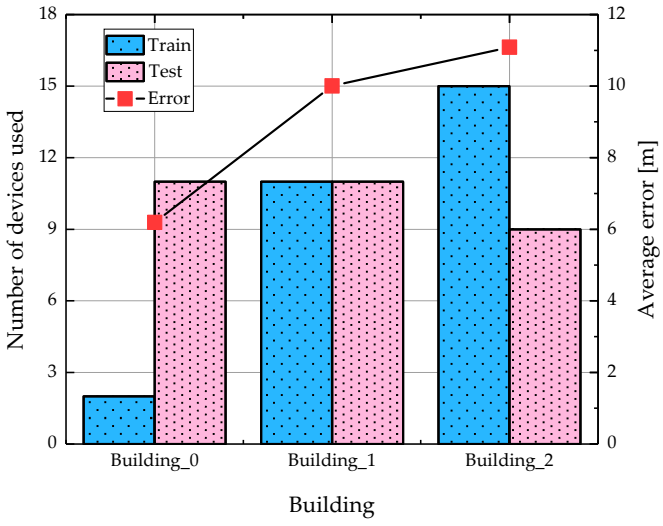

Figure 8. Relationship between the number of data acquisition devices and positioning errors.

With the rapid development of Wi-Fi protocols and hardware, the effect of the isomerism will be smaller and smaller. DNN-based Wi-Fi fingerprint location recognition shows vast potential.

4.2. Experiments of China University of Mining and Technology

4.2.1. Experimental Setting

In the UJIIndoorLoc public dataset, the test data are merely simple discrete points, which cannot be used to test the positioning accuracy of user motion. Therefore, a positioning experiment was set up on the fourth floor of the School of Environment Science and Spatial Informatics, China University of Mining and Technology, Xuzhou, China. This is a typical indoor environment with 800 square meters of the experimental area. We set up 330 fingerprint points in total (1.2 m × 1.2 m) and collected 60 Wi-Fi samples at each fingerprint point, for a total of 19,800 *RSS* training samples. During the offline stage, we observed 195 APs. The layout of the experimental site is shown in Figure 9.

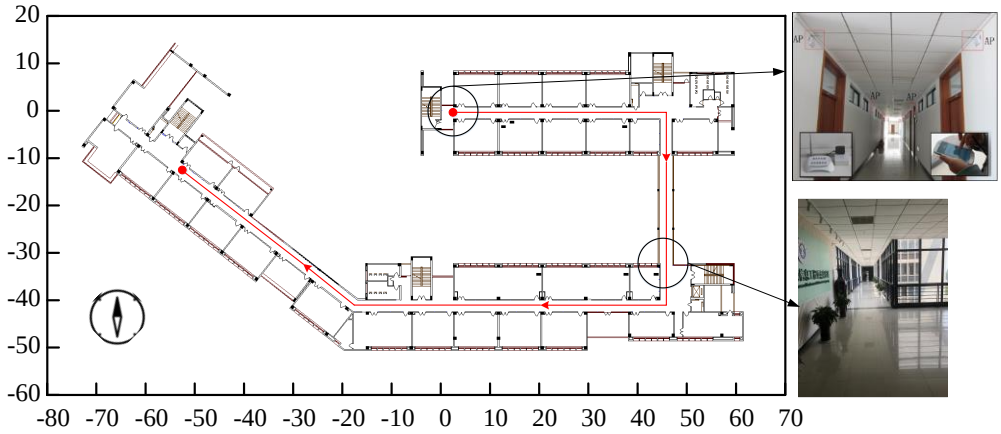

Figure 9. Layout of the experimental site.

We constructed a three-hidden-layer (400-300-200) DNN structure in line with the training data, which possessed 195 and 330 neurons in the input and output layers, respectively. The fingerprint database had 19,800 × 195 dimensions. The same smartphone models (Samsung Galaxy S5) were used as test devices to avoid the interference that heterogeneous devices can wreak on positioning results, and the sampling frequency of Wi-Fi was 0.35 Hz. The red line in Figure 9 represents the user's dynamic walking trajectory.

4.2.2. Analysis of Dynamic Positioning Results

Currently, KNN (RADAR) [16], weighted KNN (WKNN) [18] and Clustering-WKNN (CWKNN) (Horus) [17] are the gold standards of Wi-Fi positioning methods. Our evaluation of the DNN performance in dynamic Wi-Fi positioning can be seen in Figure 10, which illustrates the average error of the DNN as well as that of classic methods in dynamic motion. Further, the figure also shows the changes in the positioning accuracy of different methods two months after the establishment of the fingerprint database. Without the influence of device heterogeneity, the difference between the positioning results based on the DNN and traditional methods becomes smaller, but DNN still offers better performance. Compared with KNN, WKNN and CWKNN, the average errors of DNN are reduced as follows: 20.49%, 15.68% and 13.62%, respectively. As time goes by, the positioning errors of these methods increase to varying degrees, but the DNN still has the best positioning accuracy, with an error of 2.25 m. Figure 11 shows the average time consumption of these methods for each positioning. The DNN-based Wi-Fi positioning method has the least time consumption since there is no need to cluster and calculate the Euclidean distance of signal strength between unknown and all fingerprint points. Accordingly, experiments demonstrate that the DNN can not only offer better

positioning accuracy and better real-time performance than the traditional methods, but also have a good time robustness.

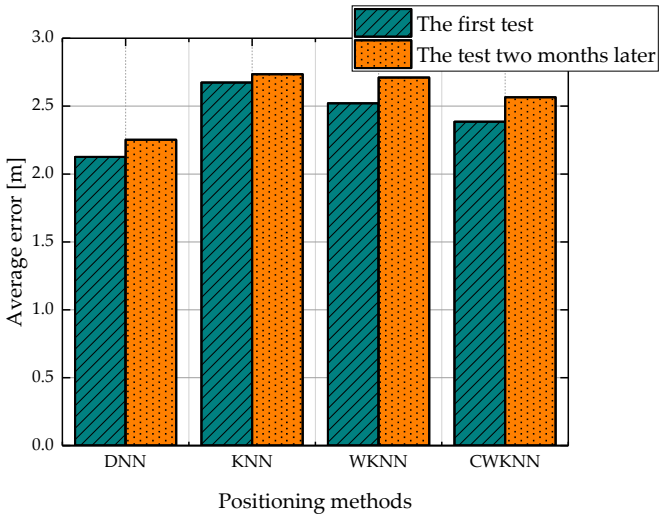

Figure 10. Positioning errors with different Wi-Fi positioning methods.

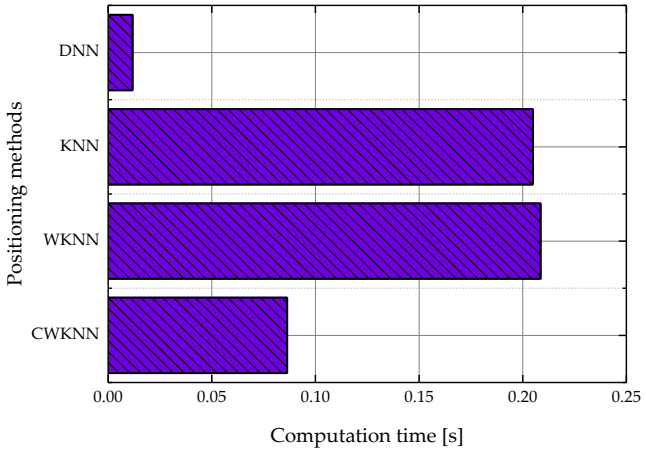

Figure 11. Average time consumption with different methods.

In order to test the effectiveness of constrained KF in dynamic positioning results, the following calculation schemes were conducted:

- Scheme 1: No use of Kalman filter.
- Scheme 2: Use of constrained Kalman filter.
- Scheme 3: Use of adaptive constrained Kalman filter.

Figure 12 shows the dynamic positioning results of the three schemes. It can be seen from the details of location trajectories that Scheme 1 without constraint inevitably suffers from typical overlapping and jumping due to the influence of multipath interference and *RSS* fluctuation. In Schemes 2 and 3, the accuracy of the positioning trajectory improved significantly with constrained KF. Further, Scheme 3 further adjusted the measurement noise adaptively to make trajectory smoother and reduce the occurrence of overlapping and jumping, so it estimated the location of the user very well. The trajectories show that the proposed adaptive constrained KF enhances the robustness of the localization results when gross errors of Wi-Fi appear.

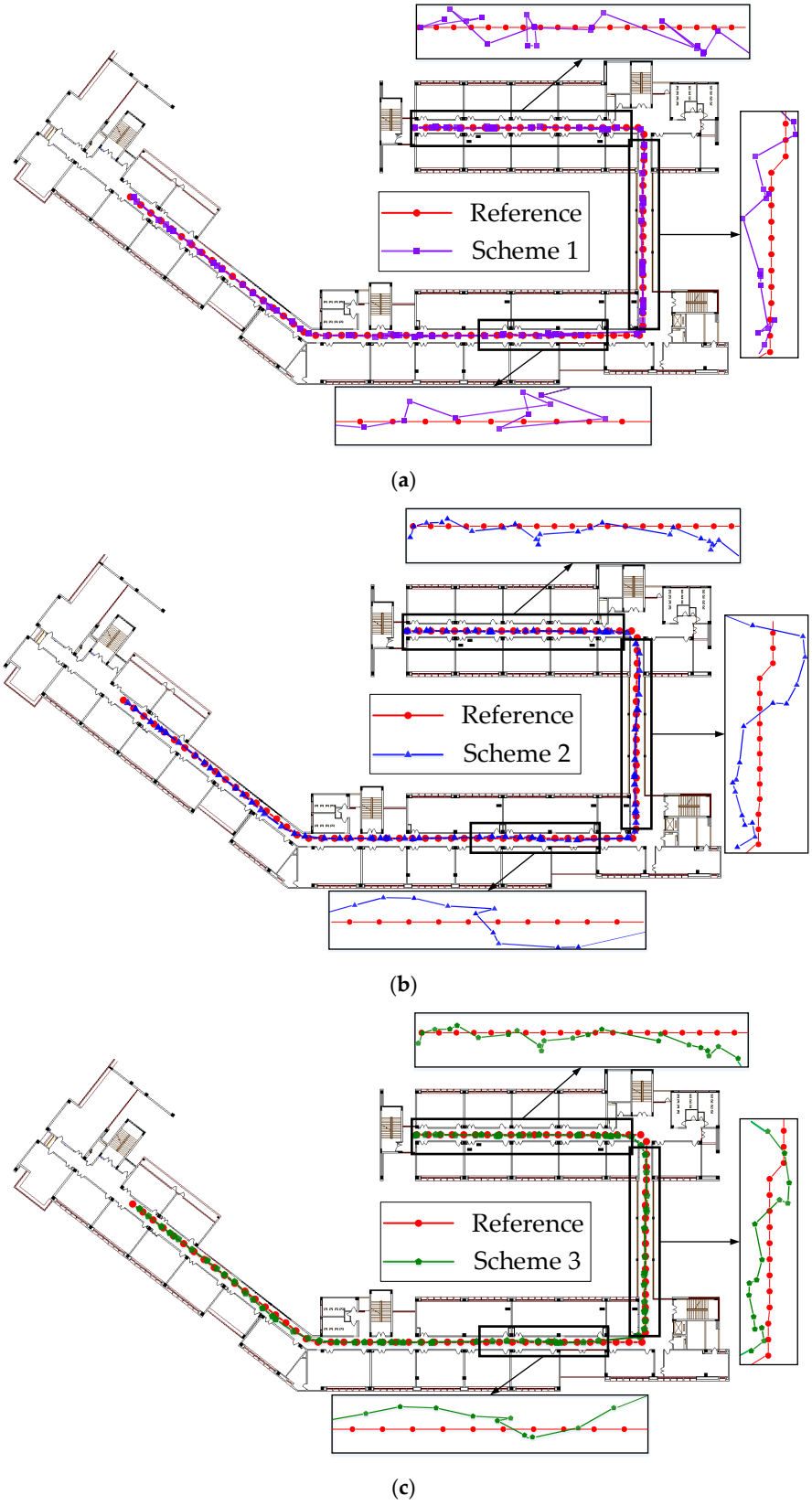

Figure 12. Positioning results of dynamic Wi-Fi positioning with different schemes: (**a**) Scheme 1; (**b**) Scheme 2; (**c**) Scheme 3.

Figures 13 and 14 show in detail the positioning errors and corresponding cumulative distribution functions (CDFs). The positioning result of Scheme 1 is not stable, and there is a large error. Schemes 2 and 3 can effectively suppress the error after using the constrained KF, and the maximum error is obviously decreased. Compared with Scheme 2, the main contribution of Scheme 3 is that the adaptive mechanism can recognize the gross error when it appears and reduce its influence on the positioning results by adjusting measurement noise. The probabilities of positioning errors of less than 3.5 m for the three schemes are 80.25%, 83.95% and 90.12% respectively.

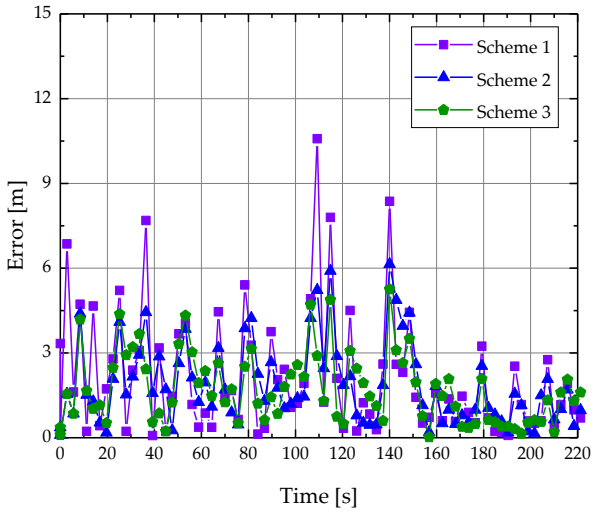

Figure 13. Positioning errors with different schemes.

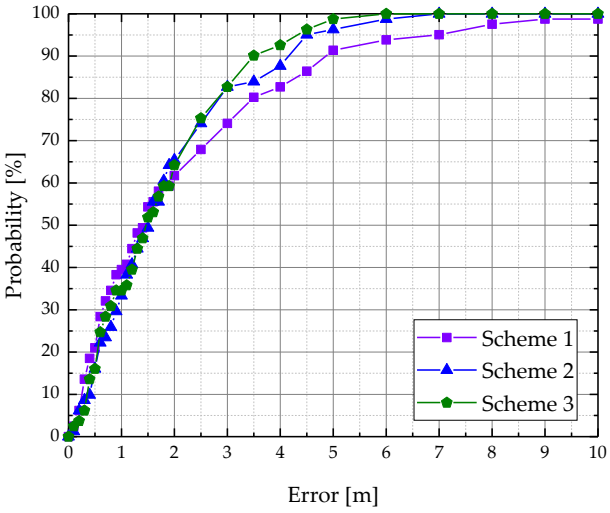

Figure 14. Cumulative distribution functions (CDFs) with different schemes.

Table 4 shows the average error, RMSE, circular-error-probability (CEP) and maximum error of the three schemes. Constrained KF improves the accuracy of dynamic positioning. Scheme 3, which uses the adaptive mechanism, offers the best performance. Compared with Schemes 1 and 2, Scheme 3 shows reduced errors as follows: (a) 19.25% and 7.03% for average error; (b) 29.47% and 8.97% for RMSE; (c) 39.07% and 5.86%for CEP; (d) 50.28% and 14.33%for maximum error. The positioning errors of Scheme 3 using adaptive constraint KF greatly reduce.

Table 4. Positioning errors for different schemes, m.

Schemes	Average Error	Room Mean Square Error (RMSE)	Circular-Error-Probability CEP (95%)	Maximum Error
Scheme 1	2.13	3.02	6.86	10.58
Scheme 2	1.85	2.34	4.44	6.14
Scheme 3	1.72	2.13	4.18	5.26

4.2.3. Analysis of Static Positioning Results

For a user who may be anywhere in an indoor environment, it is more important to determine the current position than to obtain an accurate trajectory. We utilized HMM to optimize our calculation of multiple posterior probability and obtain the optimal location estimation of the unknown point according to the temporal correlation of multiple observations on the same unknown point.

At the experimental site, we selected 39 test points at equal intervals of 5 m. Figure 15 shows the positioning situation of HMM optimal location and mean location of each test point, based on its five prior and five posterior *RSS* observations. Figure 16 and Table 5 show the error comparison between the two optimization methods. It can be seen that the results of the two methods are close when it comes to determining the real location, but HMM optimization has better performance. In addition, the positioning errors for HMM are fewer than that of using the mean. Compared with mean positions, the respective errors of HMM are reduced as follows: 0.36 m for average error, 0.46 m for RMSE, and 0.72 m for CEP. Accordingly, HMM optimization performs better than the mean when multiple observations are used to obtain accurate coordinates of the unknown point. However, it needs to perform multiple observations of the same point.

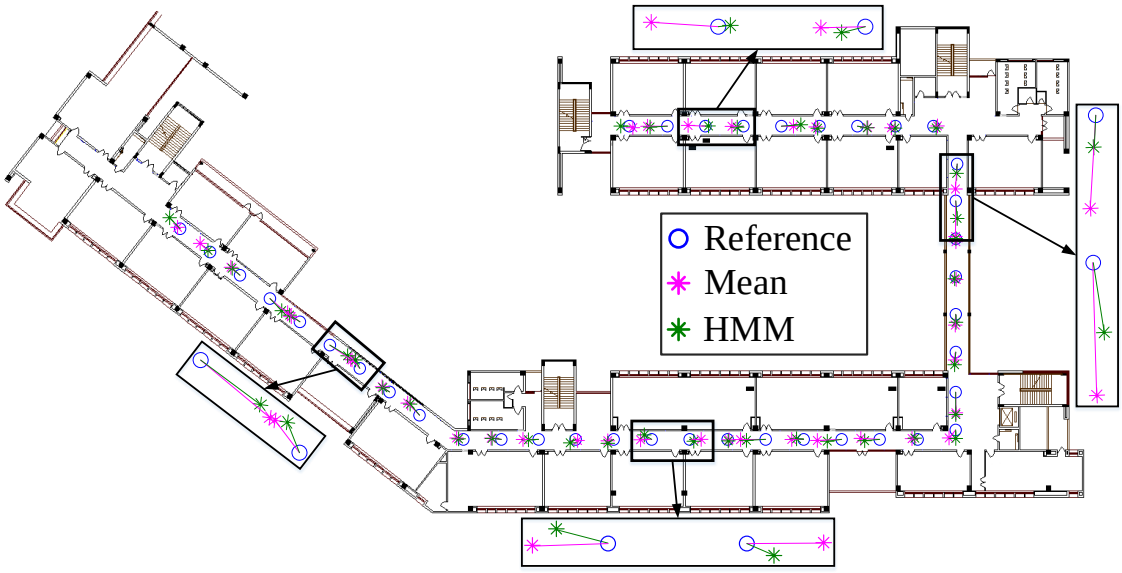

Figure 15. Positioning results calculated by mean and hidden Markov models (HMM).

Table 5. Positioning errors of mean location and HMM optimal location, m.

Algorithms	Average Error	RMSE	CEP (95%)	Maximum Error
Mean	1.58	1.89	3.18	4.52
HMM	1.22	1.43	2.46	2.84

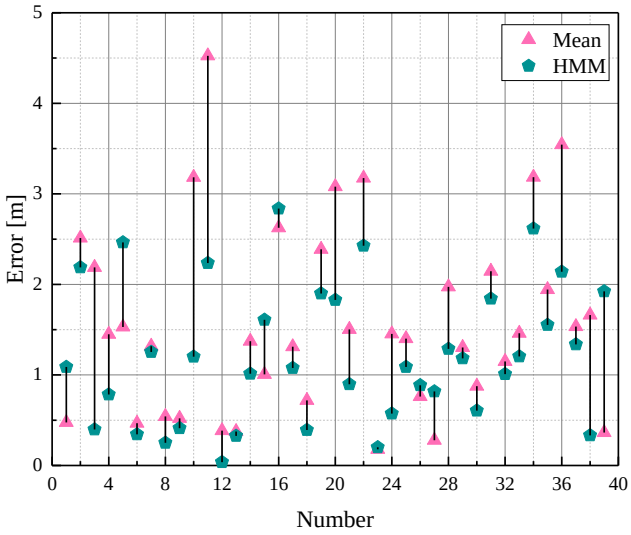

Figure 16. Positioning errors of mean and HMM locations compared with the reference.

5. Conclusions

In this paper, we propose a robust and accurate Wi-Fi fingerprint location recognition method based on a DNN. We use the adaptive constrained KF in the dynamic positioning and the HMM in the static positioning to overcome the gross error caused by the fluctuation in *RSS* and multipath interference and to improve positioning accuracy. We used a public dataset to fully verify the feasibility of DNN in Wi-Fi positioning. Experimental results show that the accuracy of DNN in floor recognition is 94.60%, and the error of position estimation is 5.29 m, which is superior to traditional methods. Further experiments in office buildings show that the positioning error of DNN is 2.13 m without interference caused using heterogeneous devices. Compared with classic methods, the DNN has higher positioning accuracy and better real-time performance. The dynamic positioning accuracy can reach 1.72 m after adaptive constrained KF smoothing. In static positioning, HMM is used to optimize the multi-observation results on the unknown points, which realizes 1.22 m positioning accuracy.

In the future, we will attempt to combine DNN with semi-supervised classification to reduce training time and update fingerprint database quickly and effectively when the environment changes, so as to ensure the accuracy of the positioning system. In the meantime, we will integrate DNN-based Wi-Fi positioning results with other sensors to build other positioning schemes based on real-world situations.

Author Contributions: Conceptualization, Y.W. and J.G.; methodology, Y.W.; software, Y.W. and Z.L.; validation, Y.W., Z.L. and L.Z.; data curation, Y.W.; writing—original draft preparation, Y.W. and J.G.; writing—review and editing, Z.L. and L.Z.; funding acquisition, J.G. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 41604006, No. 41874006, No. 41674008) and supported by Natural Science Foundation of Jiangsu Province (No. BK20160247).

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

References

1. Dos Santos, I.L.; Pirmez, L.; Lemos, É.T.; Delicato, F.C.; Pinto, L.A.V.; de Souza, J.N.; Zomaya, A.Y. A localized algorithm for Structural Health Monitoring using wireless sensor networks. *Inf. Fusion* **2014**, *15*, 114–129. [CrossRef]
2. Khalajmehrabadi, A.; Gatsis, N.; Akopian, D. Modern WLAN fingerprinting indoor positioning methods and deployment challenges. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1974–2002. [CrossRef]
3. Bordoy, J.; Wendeberg, J.; Schindelhauer, C.; Reindl, L.M. Single transceiver device-free indoor localization using ultrasound body reflections and walls. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 October 2015; pp. 1–7.

4. Ruiz, A.R.J.; Granja, F.S. Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2106–2117. [CrossRef]
5. Daroogheha, S.; Lasky, T.A.; Ravani, B. Position Measurement Under Uncertainty Using Magnetic Field Sensing. *IEEE Trans. Magn.* **2018**, *54*, 1–8. [CrossRef]
6. Zheng, Y.; Luo, P.; Chen, S.; Hao, J.; Cheng, H. Visual search based indoor localization in low light via rgb-d camera. *World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inf. Eng.* **2017**, *11*, 349–352.
7. Xie, B.; Tan, G.; He, T. Spinlight: A high accuracy and robust light positioning system for indoor applications. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, Seoul, Korea, 1–4 November 2015; pp. 211–223.
8. Harle, R. A survey of indoor inertial positioning systems for pedestrians. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1281–1293. [CrossRef]
9. Li, Q.; Li, W.; Sun, W.; Li, J.; Liu, Z. Fingerprint and assistant nodes based Wi-Fi localization in complex indoor environment. *IEEE Access* **2016**, *4*, 2993–3004. [CrossRef]
10. Gansemer, S.; Pueschel, S.; Frackowiak, R.; Hakobyan, S.; Grossmann, U. Improved RSSI-based Euclidean distance positioning algorithm for large and dynamic WLAN environments. *Int. J. Comput.* **2010**, *9*, 37–44.
11. Talvitie, J.; Renfors, M.; Lohan, E.S. Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1340–1353. [CrossRef]
12. Shaeffer, D.K. MEMS inertial sensors: A tutorial overview. *IEEE Commun. Mag.* **2013**, *51*, 100–109. [CrossRef]
13. Li, Y.; Zhuang, Y.; Zhang, P.; Lan, H.; Niu, X.; El-Sheimy, N. An improved inertial/wifi/magnetic fusion structure for indoor navigation. *Inf. Fusion* **2017**, *34*, 101–119. [CrossRef]
14. Chen, J.; Ou, G.; Peng, A.; Zheng, L.; Shi, J. An INS/WiFi indoor localization system based on the Weighted Least Squares. *Sensors* **2018**, *18*, 1458. [CrossRef] [PubMed]
15. 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 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 261–270.
16. Bahl, P.; Padmanabhan, V.N.; Bahl, V.; Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the IEEE INFOCOM 2000, Tel Aviv, Israel, 26–30 March 2000.
17. Youssef, M.A.; Agrawala, A.; Shankar, A.U. WLAN location determination via clustering and probability distributions. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, Fort Worth, TX, USA, 26 March 2003; pp. 143–150.
18. Khodayari, S.; Maleki, M.; Hamedi, E. A RSS-based fingerprinting method for positioning based on historical data. In Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Ottawa, ON, Canada, 11–14 July 2010; pp. 306–310.
19. Wu, C.; Yang, Z.; Zhou, Z.; Liu, Y.; Liu, M. Mitigating large errors in WiFi-based indoor localization for smartphones. *IEEE Trans. Veh. Technol.* **2016**, *66*, 6246–6257. [CrossRef]
20. Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Veh. Technol.* **2016**, *66*, 763–776. [CrossRef]
21. Figuera, C.; Rojo-Álvarez, J.L.; Wilby, M.; Mora-Jiménez, I.; Caamaño, A.J. Advanced support vector machines for 802.11 indoor location. *Signal Process.* **2012**, *92*, 2126–2136. [CrossRef]
22. Elbasiony, R.; Gomaa, W. WiFi localization for mobile robots based on random forests and GPLVM. In Proceedings of the 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, 3–6 December 2014; pp. 225–230.
23. Nowicki, M.; Wietrzykowski, J. Low-effort place recognition with WiFi fingerprints using deep learning. In Proceedings of the International Conference Automation, Warsaw, Poland, 15–17 March 2017; pp. 575–584.
24. Félix, G.; Siller, M.; Alvarez, E.N. A fingerprinting indoor localization algorithm based deep learning. In Proceedings of the Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 1006–1011.
25. Zhang, W.; Liu, K.; Zhang, W.; Zhang, Y.; Gu, J. Deep neural networks for wireless localization in indoor and outdoor environments. *Neurocomputing* **2016**, *194*, 279–287. [CrossRef]
26. Rizk, H.; Torki, M.; Youssef, M. CellinDeep: Robust and accurate cellular-based indoor localization via deep learning. *IEEE Sens. J.* **2018**, *19*, 2305–2312. [CrossRef]

27. Zhuang, Y.; Li, Y.; Qi, L.; Lan, H.; Yang, J.; El-Sheimy, N. A two-filter integration of MEMS sensors and WiFi fingerprinting for indoor positioning. *IEEE Sens. J.* **2016**, *16*, 5125–5126. [CrossRef]
28. Poulose, A.; Kim, J.; Han, D.S. A Sensor Fusion Framework for Indoor Localization Using Smartphone Sensors and Wi-Fi RSSI Measurements. *Appl. Sci.* **2019**, *9*, 4379. [CrossRef]
29. Li, Y.; He, Z.; Gao, Z.; Zhuang, Y.; Shi, C.; El-Sheimy, N. Toward Robust Crowdsourcing-Based Localization: A Fingerprinting Accuracy Indicator Enhanced Wireless/Magnetic/Inertial Integration Approach. *IEEE Internet Things J.* **2018**, *6*, 3585–3600. [CrossRef]
30. Bi, J.; Wang, Y.; Li, X.; Cao, H.; Qi, H.; Wang, Y. A novel method of adaptive weighted K-nearest neighbor fingerprint indoor positioning considering user's orientation. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1–13. [CrossRef]
31. Zhou, C.; Yuan, J.; Liu, H.; Qiu, J. Bluetooth indoor positioning based on RSSI and Kalman filter. *Wirel. Pers. Commun.* **2017**, *96*, 4115–4130. [CrossRef]
32. 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.
33. Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.A. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 1096–1103.
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Nevada, CA, USA, 3–6 December 2012; pp. 1097–1105.
35. Peng, Z.; Richter, P.; Leppäkoski, H.; Lohan, E.S. Analysis of crowdsensed wifi fingerprints for indoor localization. In Proceedings of the 21st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 6–10 November 2017; pp. 268–277.