

Research Article

Combination of DNN and Improved KNN for Indoor Location Fingerprinting

Peng Dai ¹, Yuan Yang,¹ Manyi Wang ², and Ruqiang Yan ^{1,3}

¹School of Instrument Science and Engineering, Southeast University, Sipailou Road 2, Nanjing 210096, China

²School of Mechanical Engineering, Nanjing University of Science and Technology, Xiaolingwei 200, Nanjing 210094, China

³School of Mechanical Engineering, Xi'an Jiaotong University, 28 Xianning West Road, Xi'an 710049, China

Correspondence should be addressed to Ruqiang Yan; yanruqiang@xjtu.edu.cn

Received 15 November 2018; Revised 18 February 2019; Accepted 21 February 2019; Published 6 March 2019

Guest Editor: Yubin Zhao

Copyright © 2019 Peng Dai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Fingerprinting based on Wi-Fi Received Signal Strength Indicator (RSSI) has been widely studied in recent years for indoor localization. While current algorithms related to RSSI Fingerprinting show a much lower accuracy than multilateration based on time of arrival or the angle of arrival techniques, they highly depend on the number of access points (APs) and fingerprinting training phase. In this paper, we present an integrated method by combining the deep neural network (DNN) with improved K-Nearest Neighbor (KNN) algorithm for indoor location fingerprinting. The improved KNN is realized by boosting the weights on K-nearest neighbors according to the number of matching access points. This will overcome the limitation of the original KNN algorithm on ignoring the influence of the neighboring points, which directly affect localization accuracy. The DNN algorithm is first used to classify the Wi-Fi RSSI Fingerprinting dataset. Then these possible locations in a certain class are also classified by the improved KNN algorithm to determine the final position. The proposed method is validated inside a room within about $13 \times 9 \text{ m}^2$. To examine its performance, the presented method has been compared with some classical algorithms, i.e., the random forest (RF) based algorithm, the KNN based algorithm, the support vector machine (SVM) based algorithm, the decision tree (DT) based algorithm, etc. Our real-world experiment results indicate that the proposed method is less dependent on the dense of access points and indoor radio propagation interference. Furthermore, our method can provide some preliminary guidelines for the design of indoor Wi-Fi test bed.

1. Introduction

Positioning technology is one of the key points for location based service (LBS). And the attention and demands on Indoor Positioning Service (IPS) increase unceasingly in recent years. For outdoor positioning, the Global Satellite Systems (GNSSs) such as American Global Positioning System (GPS) and Chinese Beidou Satellite Navigation System are performing very well. However, GNSSs are not suitable for indoor positioning as the satellite signal propagation can be easily interfered by indoor complex environment [1]. Therefore, other positioning solutions should be implemented for indoor positioning and navigation. Since wireless signal is now widely available, many approaches aiming at wireless information are proposed to estimate locations in indoor environment.

Compared with other positioning systems, Wi-Fi fingerprint positioning technology has the advantages of low-cost and high precision [2]. Due to the wide use of Wi-Fi worldwide, fingerprint positioning technology based on Wi-Fi signals can be easily constructed and put into work in indoor scenario without any additional hardware, which makes the costs reduce considerably. This technology uses Wi-Fi signal strength to model and locate, which means it is not necessary to know the exact location of the APs (Access Points). Meanwhile, the situation of signal absorption and attenuation does not need to be taken into consideration.

During the last decade, indoor positioning technology based on wireless signal has been developed rapidly with many new methods and technologies [2]. Microsoft Research Asia developed RADAR system, which is a radio-frequency-based system for locating and tracking users inside buildings

and can estimate the user location with a high degree of accuracy [3]. The University of Maryland proposed a system called Horus, and it can achieve accuracy with less than 0.6 meter on the average [4]. Moreover, Tsinghua University put forward the LiFS (Location in Fingerprint Space) system based on off-the-shelf Wi-Fi infrastructure and mobile phones [5]. As more improved algorithms are applied to indoor positioning, the results are becoming more accurate.

The location fingerprinting (LF) technique is commonly used in indoor positioning as it can take advantage of existing wireless local network (WLAN) infrastructure. To deploy a traditional LF-based indoor positioning system, location fingerprints consisted of the information of media access control (MAC) and received signal strength indicator (RSSI) of the access points (APs) should be generated in the offline phase firstly. And each coordinate corresponds to measured information of searched APs. Then, the mobile device can be localized with these fingerprints in the online phase. The measured RSSIs are compared with the location fingerprints by positioning algorithm to determine the final coordinates. Furthermore, the fingerprint database should be updated periodically to reduce the errors caused by changeable Wi-Fi environment.

There are some other issues that researchers always pay attention to, like how many APs should be required to obtain good positioning results [6]. However, the number of used APs is not the focus of this paper. We collect all observable APs existed in the positioning environment to make the fingerprint database. Then stability analysis of the APs is carried out to delete the unstable APs, which is conducive to improving accuracy. Moreover, it is known that Channel State Information (CSI, reflecting channel response in 802.11 a/g/n) has attracted many research efforts and some works have approved that it can achieve more accurate result than RSSI does [7]. However, only a few wireless network adapters can collect the physical layer feature information, such as Intel's IWL 5300. Therefore, the new technology is not able to be implemented on a smart phone unless the phone's wireless network adapter can collect CSI information. In this paper, we focus on positioning algorithm based on Wi-Fi RSSI information which is one of the key points of the indoor positioning.

Different from other approaches, a new Wi-Fi fingerprinting positioning method is proposed in this paper, which combines DNN and improved KNN to improve the accuracy of indoor positioning. Moreover, there are some contributions of this paper as follows.

- (1) The improved KNN algorithm is proposed based on the number of matched APs, which takes the relationship between the positioning neighbors into consideration.
- (2) The whole positioning scene is separated into some clusters, which increase the number of learning samples to DNN classifier algorithm. After knowing the exact cluster based on the DNN classifier, the interference of other clusters can be reduced and the computation cost of the improved KNN is decreased.

- (3) The stability of Wi-Fi APs is estimated by analyzing the quality of received signal.

The rest of the paper is structured as follows. After presenting the related work about positioning algorithms in Section 2, Section 3 introduces the proposed approach in detail. Then experimental study is carried out in Section 4, followed by analytical evaluation. Finally, Section 5 draws some conclusions.

2. Related Work

In the online phase, positioning system matches precollected data with the signal strength to determine where the user is by means of positioning algorithm. Therefore, the positioning algorithm is crucial for positioning accuracy. In this section, we will introduce some indoor positioning algorithms.

(1) *K-Nearest Neighbor*. The KNN algorithm is one of the simplest algorithms in machine learning. KNN is widely used for its low-cost and high accuracy. It compares the generated RSSI with the fingerprint data and chooses the k-nearest neighbors of fingerprint data according to the calculated distance, i.e., Manhattan distance or Euclidean distance. Therefore, the correlative coordinates of the kth position are the possible positions of users. Moreover, it can increase positioning accuracy by calculating the mean value of the k coordinates.

The radar indoor positioning system [3] applies the nearest neighbor method. Y Fang et al. [8] presented an improved KNN algorithm in fingerprinting information matching in Wi-Fi indoor positioning system. Ma et al. [9] proposed a new method called the Clustering Filtered KNN (CFK) which combined KNN with clustering.

(2) *Random Forest*. Random Forest (RF) [10] is an ensemble learning method for classification and regression. The random forest is made up of many decision trees which is set up randomly. There is no association between each decision tree in the RF. After setting up the RF, each decision tree decides which class the sample belongs to. And the final class of the sample is the maximum class processed by decision trees.

Adusumilli et al. [11] used random forest regression in the INS/GPS. In this study, the RF regression effectively modelled the highly nonlinear INS error due to its improved generalization capability. Jedari et al. [12] compared the RF with KNN and a rules-based classifier (JRip), and the results indicated that the RF classifier presents the best performance as compared to KNN and JRip classifiers with positioning accuracy higher than 91%. Mo et al. [13] proposed a coarse positioning method based on RF, which is able to customize several subregions, and test point to the region with an outstanding accuracy compared with some typical clustering algorithms.

(3) *Support Vector Machine*. The support vector machine (SVM) is one of the most practical methods in statistical learning as it translates the input space into a higher dimensional space by nonlinear transform defined by inner product

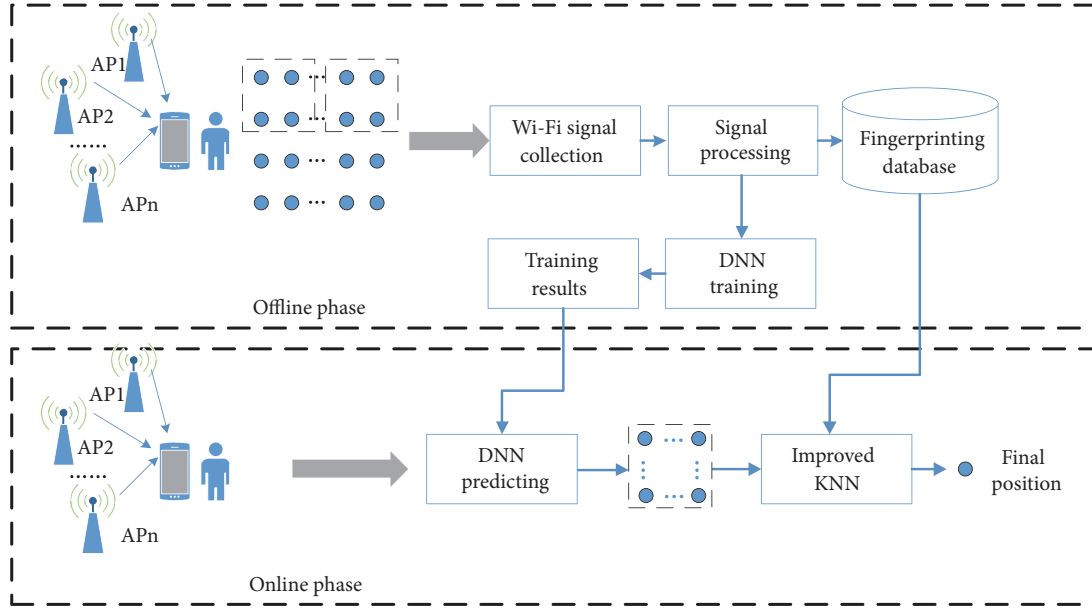


FIGURE 1: Structure of proposed algorithm.

function and calculates the optimal classification plane in this space [2]. The expression formula is shown as

$$f(x) = \text{sgn} \left(\sum_{i=1}^n T_i y_i K(x_i, x) + b \right) \quad (1)$$

Here, T_i is the Lagrange multiplier corresponding to each sample. b is the classification threshold. And $K(x_i, x)$ is the optimal classification plane of the inner product function. This function can achieve nonlinear classification after a nonlinear transformation. In addition, the SVM applied to indoor fingerprinting positioning mainly consists of the SVC problems and the SVR problems.

Yu et al. [14] utilized the information of the signal strength received from the surrounding APs to determine the user location by using SVM algorithm. They compared three kernel functions with each other; the result showed the radial function (RBF) performs the best. Figuera et al. [15] proposed a technique to enhance the SVM algorithm, which modifies the SVM algorithm to obtain three advanced methods incorporating the cross information in the two dimensions of the location.

(4) *Other Indoor Positioning Algorithms.* In addition to the above-mentioned positioning algorithms, there are many other useful algorithms applied to indoor positioning. Artificial Neural Network (ANN) is one of the most popular methods in machine learning, and many researchers take advantage of ANN methods in fingerprinting positioning, such as multilayer perceptron (MLP) and back propagation neural networks (BPNN). Shareef et al. [16] quantitatively compared the localization performance of MLP and Kalman filter, and the results showed that the MLP could potentially achieve the higher localization accuracy.

What is more, Nowicki et al. [17] employed the DNN system for building/floor classification, which achieved robust and precise classification when the sample is large enough. In addition, Ma et al. [18] proposed an improved Wi-Fi indoor positioning algorithm by weighted fusion. The algorithm used the improved Euclidean distance and the improved joint probability to calculate two intermediate results and further calculated the final result from these two intermediate results by weighted fusion.

3. The Proposed Positioning Algorithm

In this section, we introduce the positioning algorithm proposed in this paper. The procedure of the combined positioning algorithm can be divided into two phases: DNN classification phase and improved KNN classification phase. DNN algorithm is used to train the dataset in offline phase and predict in online phase, while the improved KNN algorithm classifies these promising locations in a certain class to determine the final position in online phase. And it is shown in Figure 1.

As shown in Figure 1, the proposed algorithm mainly consists of two phases: offline phase and online phase. And the offline acquisition process mainly includes four phases.

First phase is collecting indoor Wi-Fi signal. This phase collects Wi-Fi signals that are emitted by surrounding APs based on a map of collecting points. The collected signal information should contain the MAC and RSSI of the APs. Moreover, the mobile hardware can collect repeatedly at least ten times at each point.

Second phase is processing the collected signal information. And after collecting Wi-Fi signal, the data should be handled first. Unstable APs are supposed to be removed from the database in order to improve the accuracy of positioning results.

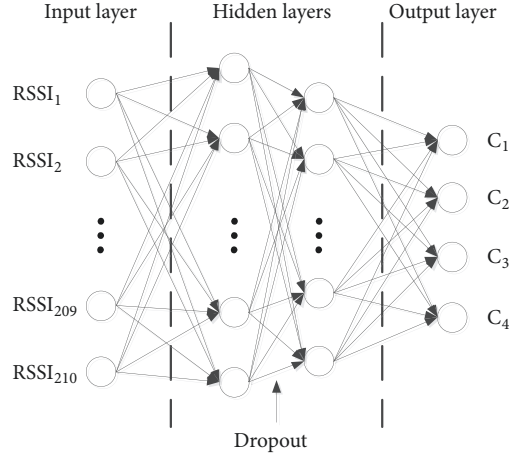


FIGURE 2: Architecture of DNN (210-256-128-4).

Then, the location fingerprinting database is constructed. And the location fingerprinting database mainly consists of the following information: MAC of APs, average value of Wi-Fi signal strength, and corresponding location.

Finally, the dataset is trained by DNN algorithm. The dataset is from the processed raw data. Firstly, the whole positioning space is divided into four main parts, which separates the dataset into four clusters. The input of the DNN algorithm is the RSSI from APs in each cluster, and the output is a certain cluster of the four clusters. The training results are stored in a file to provide for the online prediction.

In this experimental scene, the learning sample of Wi-Fi fingerprinting database is not too large; therefore, we use a simple four layers' DNN architecture which consists of two hidden layers to do the classification. The total number of APs observed in the positioning environment is about 210. Therefore, the number of neurons in the input layer is 210. Then, the two hidden layers are designed to be 256 neurons and 128 neurons, respectively. The number of neurons in output layers is equal to four. Figure 2 represents the architecture of the proposed deep neural networks. And the number in parentheses stands for the number of neurons in each layer.

W_0 , W_1 , and W_2 are defined as the weights between the RSSI values and the first hidden layer, the first and second layer, and the second and output layer, respectively. Also, b_0 , b_1 , and b_2 are defined as their biases. The first two activation functions are both Rectified Linear Unit (ReLU). The output layer is a softmax layer that outputs the probabilities of current sample belonging to analyzed classes. Therefore, the network with probabilistic generative model can be written as

$$\begin{aligned} h_1 &= \max(0, W_0 h_0 + b_0) \\ h_2 &= \max(0, W_1 h_1 + b_1) \\ V &= W_2 h_2 + b_2 \\ S_i &= \frac{e^{V_i}}{\sum_i^C e^{V_i}} \end{aligned} \quad (2)$$

where h_0 denotes the input data, i.e., the RSSI values. C is the class number. S_i represents the probability that final result belongs to the i th class.

Moreover, we employ dropout between the two hidden layers of the classifier, which randomly drops connections between the two layers during training to avoid overfitting.

The online positioning process consists of two phases as follows.

Firstly, the Wi-Fi signal information can be gathered by the positioning target. Then, the DNN algorithm can predict the cluster that the collected fingerprint belongs to.

After knowing the exact cluster from the above step, the proposed KNN algorithm is used to calculate the certain position among all possible points in the cluster.

The improved KNN algorithm is based on the traditional KNN. However, the original KNN algorithm ignores the influence of the neighboring points. The first step of the improved KNN is in accordance with the traditional one, which selects k -nearest neighbors from the certain cluster according to the average value of the squared Euclidean distance, which is expressed as

$$L_j = \frac{1}{n_j} \sum_{i=1}^{n_j} (RSS_i - RSS_j)^2 \quad (3)$$

Here, $j = 1, 2, \dots, m$ is the j th RSSI vector in fingerprint database. L_j is the average value of the squared Euclidean distance between online collected RSSI vector and the j th RSSI vector of fingerprint database. n_j is the same number of APs between online collected RSSI vector and the j th RSSI vector of fingerprint database.

After calculating the average value of the squared Euclidean distance, k -nearest neighbors are selected as the k possible target locations. Finally, certain target position is determined by the k neighbors. Moreover, each neighbor is given a weight according to the number of the matched APs. It is expressed as

$$D_{k+1} = \sum_{j=1}^k \frac{n_j}{\sum_{j=1}^k n_j} D_j \quad (4)$$

Input: RSSI
Output: Target position (P)

- (1) Get the collected RSSI vector $R(RSSI_1, RSSI_2, \dots, RSSI_n)$;
- (2) Initialize distance set D as an empty set;
- (3) Initialize weight set W as an empty set;
- (4) Initialize DNN model from the training file;
- (5) A certain cluster C that R belongs to can be predicted by DNN model;
- (6) **for** (each reference RSSI vector R_i in C) **do** //RSSI fingerprint traversal
- (7) Calculate the squared Euclidean distance D_i between R and R_i ;
- (8) Add D_i into D ;
- (9) Add the number of same APs (W_i) between R and R_i into W ;
- (10) **end for**
- (11) Select k nearest neighbors and their corresponding weights according to D ;
- (12) **for** (each position P_i in k nearest neighbors) **do** // P_i stands for coordinate of the selected position
- (13) $P = P + P_i \times W_i / \text{sum}(W_i)$; // P is the coordinate of the final position
- (14) **end for**

ALGORITHM 1: Pseudocode of online positioning process.

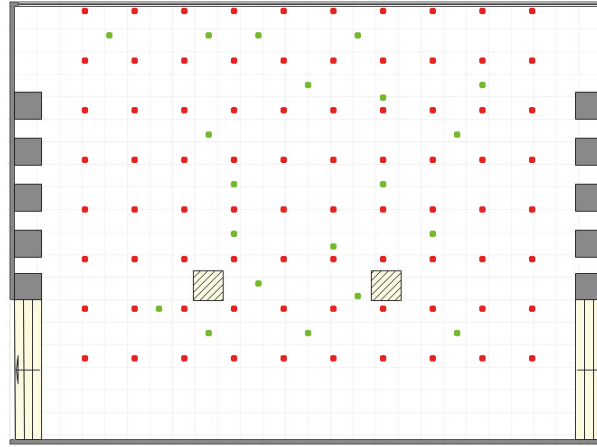


FIGURE 3: Fingerprint of the positioning environment.

Here, D_{k+1} is the final location of the mobile target. D_j is the j th possible location. n_j is the matched AP number of the j th possible location. And the pseudocode of online positioning process is shown in Algorithm 1.

The combination algorithm does not need too much training data of each point as it regards many collection points as a cluster and it trains with these clusters. Also, it can reduce the computation cost. The improved KNN algorithm only needs to calculate a certain cluster instead of all the clusters. What is more, it can improve the accuracy of positioning as it considers the relationship between the neighbors.

4. Evaluation

In this section, we carry out the experiment to validate the proposed algorithm and compare it with other algorithms, i.e., support vector machine (SVM), decision tree (DT), random forest (RF), etc.

Our experiment is carried out at the second floor in the School of Instrument Science and Engineering, SEU. The

fingerprint of the environment in our experiment is given in Figure 3.

Figure 3 shows the distribution of collecting points and experiment points, with labeled red dots and green dots, respectively, in the figure. The size of each grid is about $1.1m \times 1.1m$. The fingerprint collection is repeated 10 times at each position. Finally, we take about 20 positions in the experiment and repeat 5 times of collection at each experiment point.

In this paper, we do not add APs to structure the positioning environment; instead, we just deal with the existing APs of the environment. Thus, we should handle with some unstable APs. However, the influence of the unstable APs is not discussed in detail. We generally regard the APs whose transmitting signals are intermittent as the unstable APs. And the example of stable and unstable APs is shown in Figures 4(a) and 4(b).

Figure 4 shows the signal strength distribution of the two selected APs. Figure 4(a) shows the example of stable AP, while Figure 4(b) represents the example of unstable AP. The

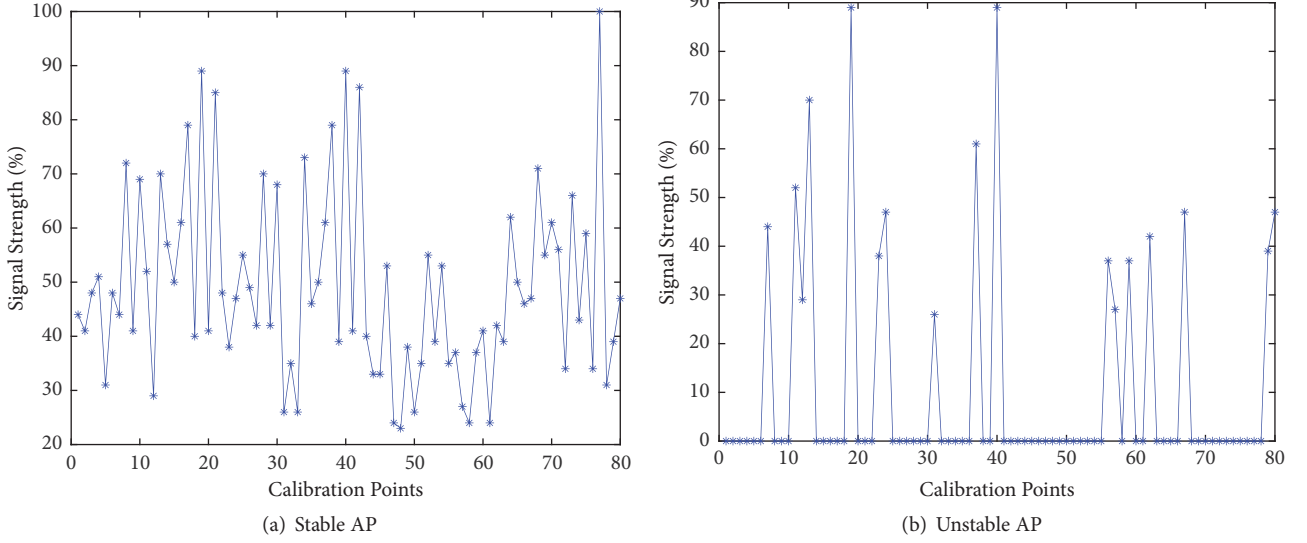
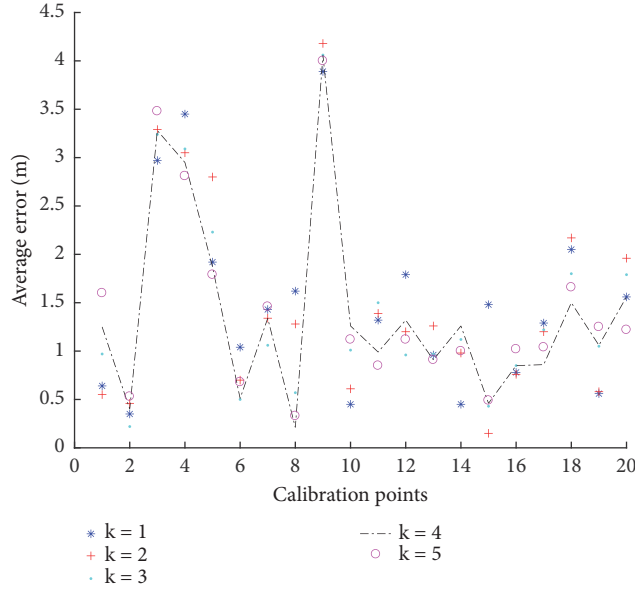


FIGURE 4: Example of stable and unstable APs.

FIGURE 5: Test result of different K values of KNN.

signal strength of AP is displayed by percentage strength ratio. And 0% of the signal strength illustrates that the tag failed to collect the AP's signal in the corresponding position.

In order to determine the algorithm efficiency, we change the value of K in KNN to make the algorithm more efficient to the positioning environment. The value of K is set from 1 to 5. The test date includes 100 collection vectors from 20 experiment points mentioned above. The positioning errors are based on the Euclidean distance from the real coordinates, which can be expressed as

$$Dis_err = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (5)$$

Here, Dis_err is the Euclidean distance between the real location (x_0, y_0) and the estimated location (x, y) .

After doing average filtering in each location, the average error in each testing location is calculated to evaluate the results of different K values. The result is given in Figure 5.

Table 1 presents test result comparison among different K values, which gives the average error of testing.

From Figure 5 and Table 1 we can conclude that when $K=4$, it performs better than other K values, where the errors of 40% experiment points are less than $1m$ and 85% are less than $2m$. Moreover, the average error of the 20 test points is $1.39m$. Therefore, we take 4 as the K -value to achieve a better performance.

Then we calculate the error of each measurement without using an average filter, and the result is shown by Figure 6, it shows all the errors of the 100 times positioning within repeating 5 times measurements at each of the 20 locations.

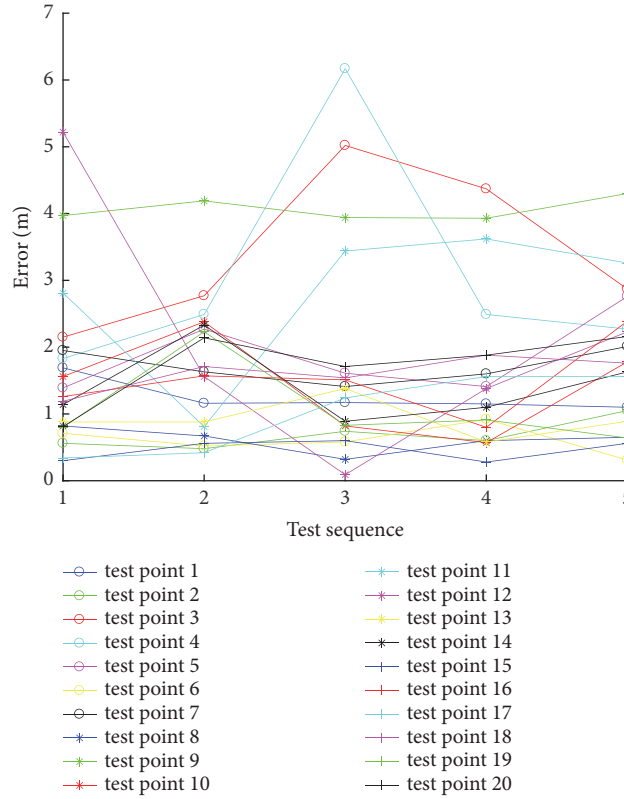


FIGURE 6: Test error of each location.

TABLE 1: Average error of different K values.

K -value	Average error (m)
1	1.50
2	1.50
3	1.43
4	1.39
5	1.42

We can figure out most positioning errors are less than $3m$. After calculation, the average positioning error is about $1.67m$.

Moreover, we compare the proposed algorithm with other classical positioning algorithms from two aspects: before average filtering and after average filtering. The result is shown by Figure 7, where the abscissa is the abbreviations of algorithms for the proposed method, Decision Tree, Gaussian Naïve Bayes, K-Nearest Neighbor, Deep Neural Networks, Support Vector Machine, and Random Forest. We can conclude from Figure 7 that the three algorithms of the best performance are TPM, SVM, and RF. Therefore, we compare the probability of the three algorithms within different Dis_err of 100 positioning as shown in Figure 8. For example, as for the proposed method, if the Dis_err is within $1m$, the probability of it is about 0.45.

From Figures 7 and 8, we can lead to the following conclusions:

- (1) The average filter can get good results. It improves the accuracy of all tested algorithms.
- (2) The proposed algorithm performs the best among the traditional algorithms.
- (3) By comparing the three algorithms with best performance, we can conclude that the proposed algorithm combines the advantages of these two traditional algorithms. As a result, the accuracy of the proposed algorithm is higher.

To sum up, the reasons why the proposed algorithm performs better are as follows:

- (1) The improved KNN algorithm could improve the accuracy of the traditional KNN algorithm in positioning.
- (2) The DNN algorithm is used to determine the most likely cluster which the target belongs to. It can not only decrease the computation cost of the improved KNN, but also reduce the interference of other clusters.
- (3) The proposed algorithm combines both advantages of the two algorithms and thus achieves better accuracy than the two traditional algorithms.

5. Conclusions

Wi-Fi indoor positioning depends on the Wi-Fi signal to get indoor location information, which is of great use and significance to the indoor positioning application. In this

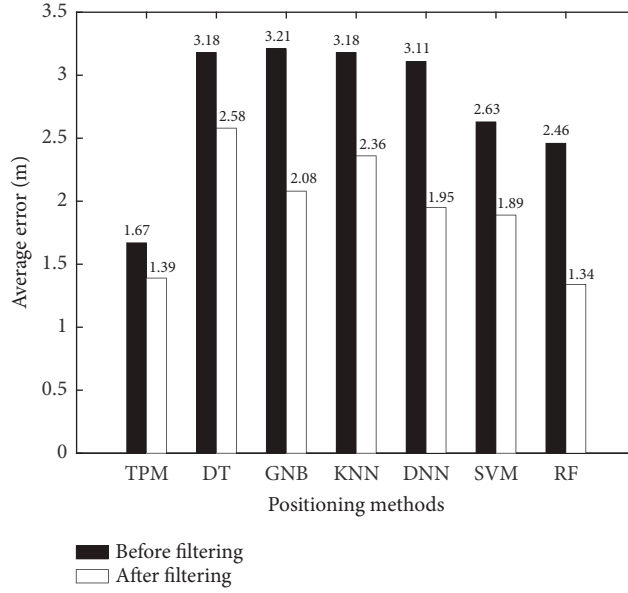


FIGURE 7: The compare results of algorithms.

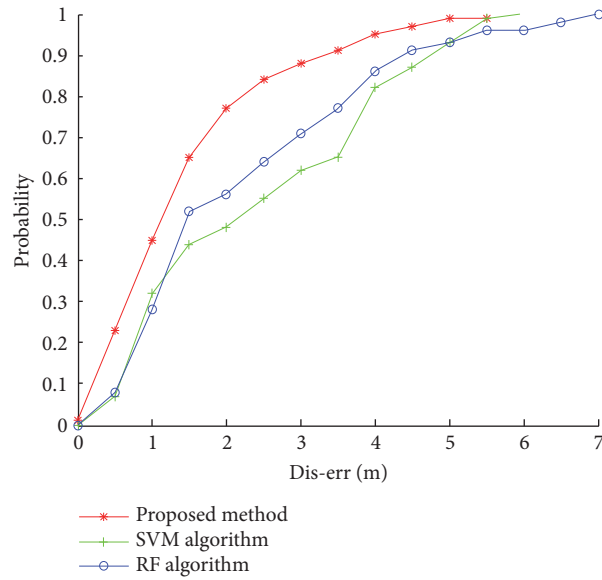


FIGURE 8: Probability distribution of 100 positioning.

paper, we mainly focus on the improvement of positioning algorithm. Firstly, we improve the KNN algorithm by considering the number of matching APs. Then we combine the DNN algorithm with the improved KNN algorithm. Moreover, in order to show the good performance of the proposed algorithm, we compare it with some traditional indoor positioning algorithms, and the result shows our proposed algorithm has a better performance.

However, it takes much manpower and time to maintain the Wi-Fi fingerprinting database in this research. Therefore, in further work, we will concentrate on obtaining the Wi-Fi fingerprinting database by simulating the wireless propagation model and a self-adapting system for replying the

change of the radio map and solving the problem of human interference.

Data Availability

The Wi-Fi fingerprinting raw data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangsu Province of China (Grant no. BK20160696), the projects of National Natural Science Foundation of China (Grant no. 61601123), and the Youth Science Fund of Jiangsu Province (Grant no. BK20160841).

References

- [1] B. Li, J. Zhang, A. G. Dempster, and C. Rizos, "Open source GNSS reference server for Assisted-Global Navigation Satellite Systems," *Journal of Navigation*, vol. 64, no. 1, pp. 127–139, 2011.
- [2] S. Xia, Y. Liu, G. Yuan, M. Zhu, and Z. Wang, "Indoor fingerprint positioning based on Wi-Fi: an overview," *ISPRS International Journal of Geo-Information*, vol. 6, no. 5, p. 135, 2017.
- [3] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.
- [4] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 205–218, ACM, June 2005.
- [5] C. Wu, Z. Yang, and Y. Liu, "Smartphones based crowdsourcing for indoor localization," *IEEE Transactions on Mobile Computing*, 2014.
- [6] P. Jiang, Y. Zhang, W. Fu, H. Liu, and X. Su, "Indoor mobile localization based on Wi-Fi fingerprint's important access point," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 429104, 8 pages, 2015.
- [7] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: indoor localization via channel response," *ACM Computing Surveys*, vol. 46, no. 2, article 25, 2013.
- [8] Y. Fang, Z. Deng, and C. Xue, "Application of an improved K nearest neighbor algorithm in WiFi indoor positioning," in *China Satellite Navigation Conference (CSNC) 2015 Proceedings: Volume III*, vol. 342 of *Lecture Notes in Electrical Engineering*, pp. 517–524, Springer, Berlin, Germany, 2015.
- [9] J. Ma, X. Li, X. Tao, and J. Lu, "Cluster filtered KNN: A WLAN-based indoor positioning scheme," in *Proceedings of the 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, pp. 1–8, Newport Beach, Calif, USA, June 2008.
- [10] T. K. Ho, "Random decision forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278–282, Montreal, Que., Canada, 1995.
- [11] S. Adusumilli, D. Bhatt, H. Wang, P. Bhattacharya, and V. Devabhaktuni, "A low-cost INS/GPS integration methodology based on random forest regression," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4653–4659, 2013.
- [12] E. Jedari, Z. Wu, R. Rashidzadeh, and M. Saif, "Wi-Fi based indoor location positioning employing random forest classifier," in *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015*, Canada, October 2015.
- [13] Y. Mo, Z. Zhang, Y. Lu, W. Meng, and G. Agha, "Random Forest Based Coarse Locating and KPCA Feature Extraction for Indoor Positioning System," *Mathematical Problems in Engineering*, vol. 2014, Article ID 850926, 8 pages, 2014.
- [14] F. Yu, M. H. Jiang, J. Liang et al., "An indoor localization of WiFi based on support vector machines," *Advanced Materials Research*, vol. 926-930, pp. 2438–2441, 2014.
- [15] C. Figuera, J. L. Rojo-Álvarez, M. Wilby, I. Mora-Jiménez, and A. J. Caamaño, "Advanced support vector machines for 802.11 indoor location," *Signal Processing*, vol. 92, no. 9, pp. 2126–2136, 2012.
- [16] A. Shareef, Y. Zhu, M. Musavi, and B. Shen, "Comparison of MLP neural network and kalman filter for localization in wireless sensor networks," in *Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems*, pp. 323–330, USA, November 2007.
- [17] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with WiFi fingerprints using deep learning," *Advances in Intelligent Systems and Computing*, vol. 550, pp. 575–584, 2017.
- [18] R. Ma, Q. Guo, C. Hu, and J. Xue, "An improved WiFi indoor positioning algorithm by weighted fusion," *Sensors*, vol. 15, no. 9, pp. 21824–21843, 2015.

