Cao *et al. Satellite Navigation* (2021) 2:27
https://doi.org/10.1186/s43020-021-00058-8

Satellite Navigation
https://satellite-navigation.springeropen.com/

ORIGINAL ARTICLE Open Access

A universal Wi-Fi fingerprint localization method based on machine learning and sample differences

Xiaoxiang Cao, Yuan Zhuang*, Xiansheng Yang, Xiao Sun and Xuan Wang

Abstract

Wi-Fi technology has become an important candidate for localization due to its low cost and no need of additional installation. The Wi-Fi fingerprint-based positioning is widely used because of its ready hardware and acceptable accuracy, especially with the current fingerprint localization algorithms based on Machine Learning (ML) and Deep Learning (DL). However, there exists two challenges. Firstly, the traditional ML methods train a specific classification model for each scene; therefore, it is hard to deploy and manage it on the cloud. Secondly, it is difficult to train an effective multi-classification model by using a small number of fingerprint samples. To solve these two problems, a novel binary classification model based on the samples' differences is proposed in this paper. We divide the raw fingerprint pairs into positive and negative samples based on each pair's distance. New relative features (e.g., sort features) are introduced to replace the traditional pair features which use the Media Access Control (MAC) address and Received Signal Strength (RSS). Finally, the boosting algorithm is used to train the classification model. The UJIndoorLoc dataset including the data from three different buildings is used to evaluate our proposed method. The preliminary results show that the floor success detection rate of the proposed method can reach 99.54% (eXtreme Gradient Boosting, XGBoost) and 99.22% (Gradient Boosting Decision Tree, GBDT), and the positioning error can reach 3.460 m (XGBoost) and 4.022 m (GBDT). Another important advantage of the proposed algorithm is that the model trained by one building's data can be well applied to another building, which shows strong generalizable ability.

Keywords: Fingerprint-based positioning, Sample difference, Binary-classification, Boosting, Machine learning, Wi-Fi positioning

Introduction

The outdoor location service has increasingly matured with the rapid development of the Global Navigation Satellite System (GNSS) (Liu et al., 2020). However, GNSS fails to provide indoor positioning service due to its signal obstruction and attenuation. While indoor positioning has become more and more important in people's daily activities, such as shopping, parking, and health monitoring. Accordingly, many scholars have conducted considerable research on indoor positioning with various techniques, such as Wi-Fi, Bluetooth, geomagnetic localization, Radio Frequency Identification (RFID), ultra-wideband, wireless local area network, computer vision, light visible communication, and Pedestrian Dead Reckoning (PDR) assisted by accelerator and gyroscope (He & Chan, 2016; Naser and Li, 2021; Zhuang et al., 2018; Yang et al., 2015; El-Sheimy & Li, 2021; El-Sheimy & Youssef, 2020).

Among these techniques, Wi-Fi positioning has become a research hotspot due to its mature hardware and software ecology, low cost, and no need of extra deployment. Main Wi-Fi positioning algorithms include Access Point (AP) proximity-aware (Hodes et al., 1997), fingerprint-based positioning (Zhuang et al., 2016), and trilateration localization based on the signal propagation

*Correspondence: yuan.zhuang@whu.edu.cn
State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China

model (Bahl & Padmanabhan, 2000). But the fingerprinting algorithm is more widely used because it can achieve the highest positioning accuracy.

Currently, the neighbor point mismatch is a prime problem in Wi-Fi fingerprint-based positioning. The traditional solution calculates the similarity between the fingerprint RSS vector and the observation RSS vector using different indices, like the Euclidean distance (Kaemarungsi & Krishnamurthy, 2004), cosine similarity (Han et al., 2015), Pearson coefficient (Li et al., 2019), and others (Machaj et al., 2011). Most of these methods use the direct differential computation method by the means of RSS vectors. However, it is difficult to describe the complex nonlinear relationship between signal vectors accurately. Therefore, many scholars recently use Machine Learning (ML) and Deep Learning (DL) for neighbor point matching. It can be broadly divided into two groups One is the supervised learning methods which use various classification methods, like Random Forest (RF) (Lee et al., 2019), Decision Tree (DT) (Chanama & Wongwirat, 2018), Bayes (Chen et al., 2013), Support Vector Machine (SVM), Neural Network (NN) (Zhang et al., 2013; Esmond & Bernard, 2013), Convolutional Neural Network (CNN) (Shao et al., 2018) and other classification algorithms (Feng et al., 2014; Li et al., 2021). The other is unsupervised learning using the methods of clustering, K-Means (Chen et al., 2015), fuzzy cluster (Bi et al. 2018), Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Deng et al., 2018), etc.

These two groups of methods have their obvious weaknesses and strengths. The classification algorithms always require a high demand, which includes both sample quality and sample quantity. Considering the time and labor costs, we can easily find that classification, especially the multi-classification, may not be suitable for fingerprint-based positioning since the classifier training requires each of these categories has a large number of samples. Therefore, many methods for sample enhancement have been proposed, for example, crowdsourced data collection (Guo & Pun, 2019), interpolation methods for sample creation (Kolakowski, 2020), and the DL to increase the size of samples, in which the most common method is the Generative Adversarial Network (GAN) (Liu & Wang, 2020; Zou et al., 2020). But the data generated by this method has poor quality, and the generation model is hard to converge when using GAN.

The clustering algorithm also has some problems. Firstly, the computational complexity of the clustering is too high to be used in real-time positioning. Secondly, clustering is more applicable for zone localization, and the accuracy of the point localization using this algorithm is always low. In addition, most of the clustering algorithms require a known number of classes and some initial centers of the clusters, which makes it hard for practical use. The abnormal data has a greater effect on the final result when compared with other methods.

The above ML-based or DL-based methods all face the same problem. They use the APs' RSS values as the input features, but the RSS values have a strong relationship with the location of the fingerprint. The classifier trained by the fingerprint data of one building cannot be used in another building, sometimes even on another floor. It requires that each building or floor trains and manages its own classifier, which can cause some problems. The first and foremost problem is the model deployment in the servers for practical application. It is necessary to deploy a huge number of models and update the models periodically, which is costly. Another problem is the model management if there are many models on the cloud. It is hard to maintain effectively, and also requires countless resources for the operation of the whole cloud platform.

To solve the above problems, a novel method is proposed using the differences among the samples. To make full use of the differences, we adopt the relative features, like the repeated AP, the signal similarity, and the other features rather than the commonly used absolute features. The boosting algorithms of the eXtreme Gradient Boosting (XGBoost) and the Gradient Boosting Decision Tree (GBDT) are used in this paper for binary classification model training rather than the multi-classification, because they are widely used in binary classification and their performance is much better than others. The test datasets perform well by using the classifier trained by the same building's data, or another building's data.

System components and methodology

System components

Figure 1 shows the proposed positioning system that involves two main phases, i.e., offline and online.

In the offline phase, the main work is the fingerprint collection and the classification model training. To improve the quality of the samples, it is better to collect RSS values serval times in each fingerprint. After the RSS values are collected, all the samples are paired. If the two samples of one pair come from the same points or neighbor points, these pairs are regarded as positive pairs, like FP_{A1} and FP_{A2} which are collected in Point A in Fig. 1. If they come from different points and the distance between them is large enough, they belong to a negative pair, like FP_{A1} which is collected in Point A and FP_{B1} which is collected in Point B in Fig. 1. We choose the FPs that come from different fingerprint points as a negative pair in this paper. Then, new features which represent the difference between two samples in each pair are calculated. The new extracted features are the inputs of the classifier rather than the original MAC-RSS pairs. The boosting algorithm

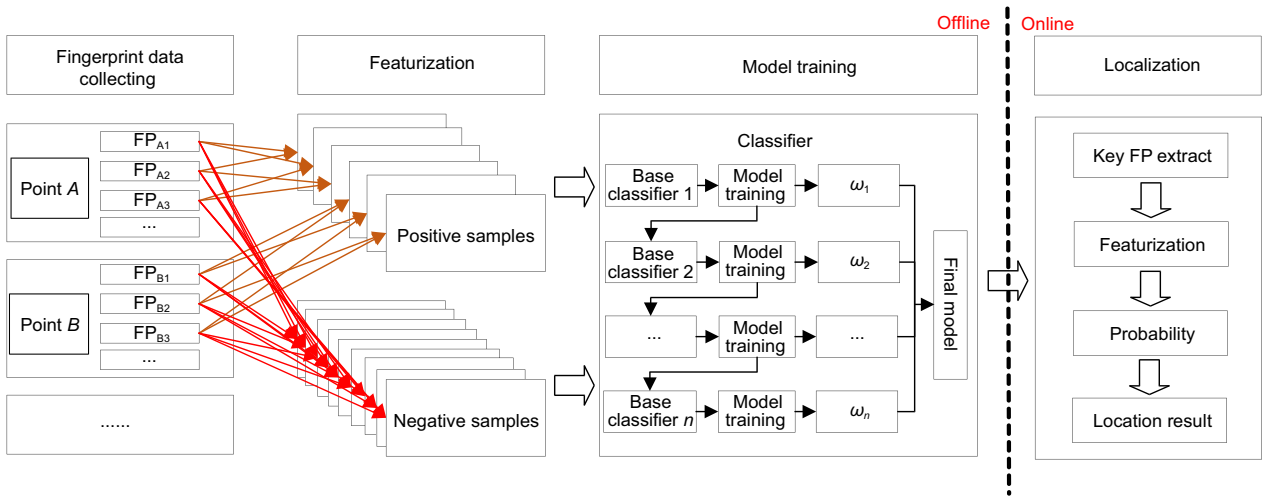

Fig. 1 The components of the whole system

is used for classifier training. Some base classifiers are used for the training process, and the output of the previous classifier works as the input of the next classifier. The misclassified samples will be considered more in the next classification. A binary classification model will be the output at the end. The output of the model is the probability with which the observation and the fingerprint come from the same point or neighbor points.

In the positioning phase, some key fingerprint points which have the common MAC with the observation list are selected for the next calculation. Then, the features from these fingerprints and observation data are calculated. The features are input into the model trained in the offline phase. And the model will output the probability of each fingerprint point and the attribute (neighbor point or not) of each fingerprint point. Finally, the point which holds highest probability is the final localization result.

Feature selection

The traditional fingerprinting methods always use the MAC-RSS pairs as the features. This means the features have a strong relationship with the locations of the fingerprint points, which limits the use of the multiple classifiers. We proposed a method to use positive and negative pairs for classification. If the two samples of one pair come from the same point or neighbor points, these pairs are regarded as positive pairs. If they come from different points and their spacing is large enough, they are negative pairs, which is shown in Fig. 2. Then, the features which represent the difference between two samples in each pair are calculated.

The new features are divided into four types, i.e., the coincidence number features, the sort feature, the similarity feature, and the transposition feature. All these features are listed in Fig. 3.

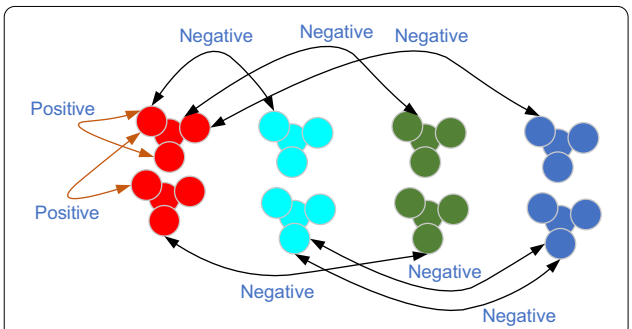

Fig. 2 The examples of positive sample and negative sample

Figure 4 and the following equations show how to calculate these features.

The similarity features can be calculated by the following equations:

a. Euclidean Distance:

$$D_e = \sqrt{\frac{\sum_{i=1}^{m}\left(rss_{(ap_i inSA)} - rss_{(ap_i inSB)}\right)^2}{m}} \quad (1)$$

where $rss_{(ap_i inSA)}$ and $rss_{(ap_i inSB)}$ represent the RSS value of ap_i in scanlist SA and SB respectively. m is the repeated MAC number of scanlist SA and SB.

b. Cosine Similarity:

$$D_{cos} = \frac{\sum_{i=1}^{m} rss_{(ap_i\, in\, SA)} * rss_{(ap_i\, in\, SB)}}{\sqrt{\sum_{i=1}^{m} rss^2_{(ap_i\, in\, SA)} \sum_{i=1}^{m} rss^2_{(ap_i\, in\, SB)}}} \quad (2)$$

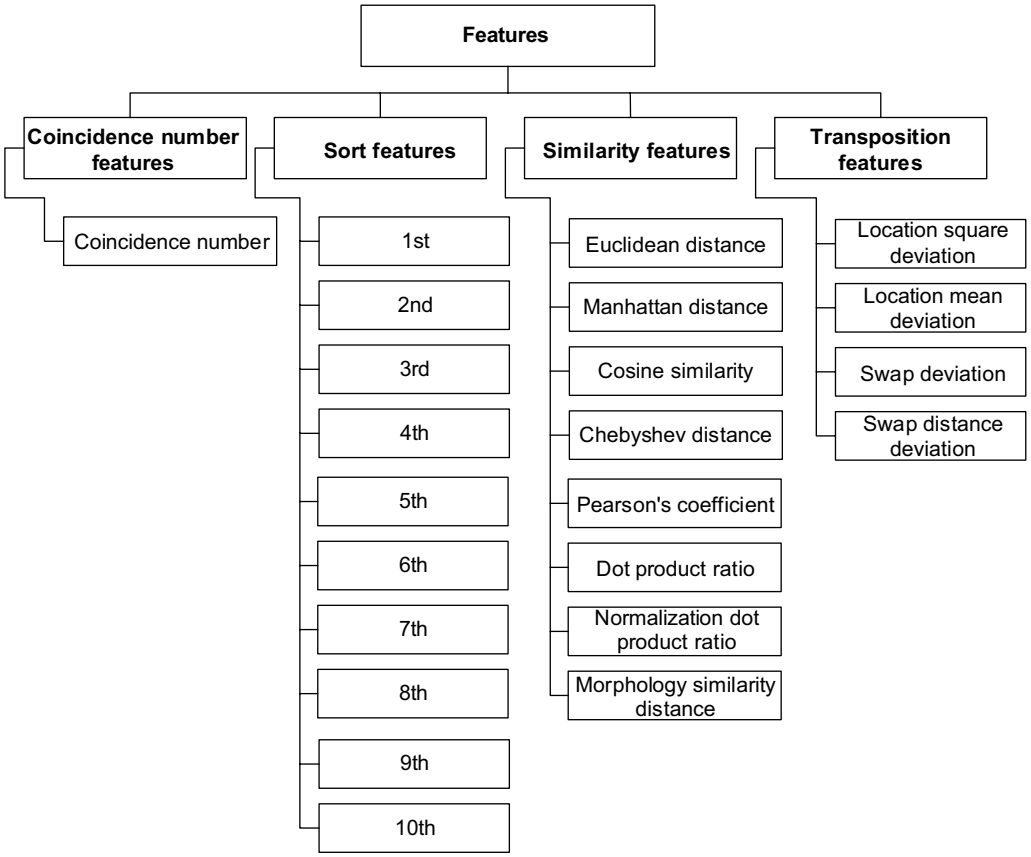

Fig. 3 The new features extracted from fingerprints

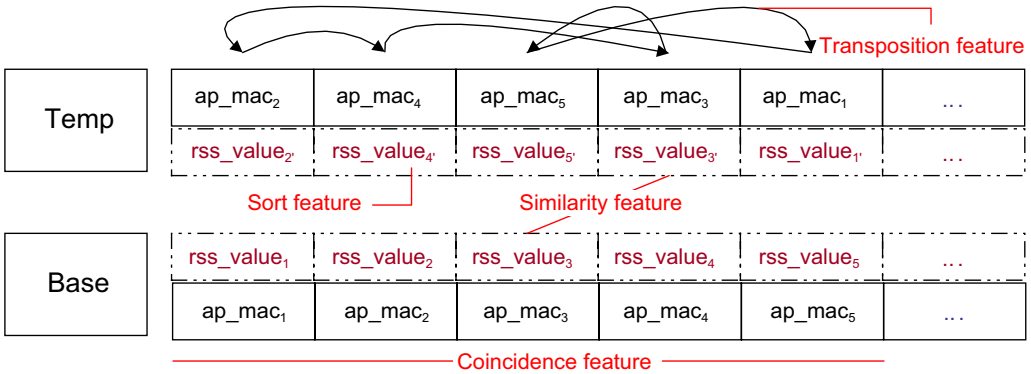

Fig. 4 The feature calculation

c. Chebyshev Distance:

$$D_Q = \mathrm{Max}\left(\left|rss_{(ap_i inSA)} - rss_{(ap_i inSB)}\right|\right), \quad i = 1, 2, 3, \ldots, m \tag{3}$$

where $\mathrm{Max}(\cdot)$ denotes the maximum function.

d. Pearson's Coefficient:

$$D_P = \frac{\sum_{i=1}^{m}\left(rss_{(ap_i inSA)} - \overline{rss_{SA}}\right)\left(rss_{(ap_i inSB)} - \overline{rss_{SB}}\right)}{\sqrt{\sum_{i=1}^{m}\left(rss_{(ap_i inSA)} - \overline{rss_{SA}}\right)^2 \sum_{i=1}^{m}\left(rss_{(ap_i inSB)} - \overline{rss_{SB}}\right)^2}} \tag{4}$$

where $\overline{rss_{SA}}$ and $\overline{rss_{SB}}$ are the means of the RSS values of scanlist *SA* and *SB*.

e. Manhattan Distance:

$$D_m = \frac{\sum_{i=1}^{m} \left| rss_{(ap_i inSA)} - rss_{(ap_i\, in\, SB)} \right|}{m} \tag{5}$$

f. Dot Product Ratio (DPR):

$$D_{dpr} = \frac{\sum_{i=1}^{m} rss_{(ap_i inSA)} * rss_{(ap_i inSB)}}{\sum_{i=1}^{m} RSS^2_{(ap_i inSA)}} \tag{6}$$

g. Normalization Dot Product Ratio (NDPR):

$$D_{ndpr} = \frac{2m-1}{m+1} D_{DPR} - \frac{m-2}{m+1} \tag{7}$$

h. Morphological Similarity Distance:

$$D_{msd} = \sum_{i=1}^{m} \sqrt{\left(rss_{(ap_i inSA)} - rss_{(ap_i inSB)}\right)^2} \times \left(2 - \frac{\left|\sum_{i=1}^{m} \left(rss_{(ap_i inSA)} - rss_{(ap_i inSB)}\right)\right|}{\sum_{i=1}^{m} \left|rss_{(ap_i inSA)} - rss_{(ap_i inSB)}\right|}\right) \tag{8}$$

The transposition feature can be calculated by the following formulas:

a. Location Square Deviation (LSD)

$$D_{lsd} = \frac{1}{N} \sum_{ap_{i=0}}^{ap_{N-1}} (ap_i[SA] - ap_i[SB])^2 \tag{9}$$

where *N* represents the length of the AP scanlist, and *SA*, *SB* are the different AP scanlists. $ap_i[SA]$ denotes the sort of ap_i in A, $ap_i[SB]$ has the same meaning as $ap_i[SA]$.

b. Location Mean Deviation (LMD):

$$D_{lmd} = \frac{1}{N} \sum_{ap_{i=0}}^{ap_{N-1}} |ap_i[A] - ap_i[B]| \tag{10}$$

c. Swap Deviation (SD):

$$D_{sd} = \mathrm{Min}(W(A \to B)) = Min\left(\sum_{s \in A \to B} 1\right) \tag{11}$$

where $SA \to SB$ represents the sift operation from *SA* to *SB*, $W(\cdot)$ is the weight function, and $\mathrm{Min}(\cdot)$ is the minimum function.

d. Swap Distance Deviation (SDD):

$$D_{sdd} = \mathrm{Min}(W(SA \to SB)) = \mathrm{Min}\left(\sum_{s \in SA \to SB} |i-j|\right) \tag{12}$$

where *i*, *j* represents one AP's sort in AP scanlist *SA* and *SB*, respectively.

Additional processes

Removal of unreliable APs

There are many mobile phones or other mobile devices in the building, and they may have a great influence on fingerprint information collecting and online positioning. In addition, the proposed method requires the ordering information. It is necessary to delete these mobile APs firstly.

The mobile MAC in fingerprint data is easy to be found and deleted by the statistical means, like the number of repeated occurrences, the cover area, the RSS values and so on.

It is hard to delete the mobile MAC in online positioning due to the limited information in the observation. There are two general solutions. The first one is comparing the current observation with the historical observation. Some abnormal MAC can be found. The second solution requires the system to maintain an abnormal field list to detect the abnormal MAC in real time. The abnormal field list may contain some obvious abnormal fields, like 'Mobile', 'HUAWEI', 'OPPO', 'VIVO', 'XIAOMI', 'smartphone', and so on.

RSS normalization

The RSS received by different types of devices is different because of the device heterogeneity. Thus, we must map them to a uniform range using Eq. (13) (Song and Wang 2017), which can mitigate the impact of the device heterogeneity to some extent.

$$rss'_{ap_i} = \frac{rss_{ap_i} - \overline{rss}}{rss_{std}} \tag{13}$$

where $\overline{rss}$ and rss_{std} respectively represents the mean value and the standard value of the RSS list.

Feature enhancement

The total number of the new features is 23, which is limited for model training. Except for the whole RSS

Table 1 The detailed data information for each building

Building ID	Floors	Training samples	Testing samples	All samples
Building0	4	4199	1050	5249
Building1	4	4157	1039	5196
Building2	5	7594	1898	9492

Table 2 The number of MAC repetitions between each building

Building ID	Building0	Building1	Building2
Building0	199	59	7
Building1	59	207	82
Building2	7	82	203

list, we also choose the top three list and the top five list to calculate these features respectively, and then the total number of the features is increased to 57. In the real test, the added features can improve the accuracy. However, some features are strongly correlated; therefore, a part of the useless features should be dropped in the next experiment.

Experiment and result discussion

Data declaration

The UJIndoorLoc Dataset (Torres-Sospedra et al. 2014) collection covers an area of 108,703 m^2, including three buildings with 3–5 floors. The Wi-Fi data are collected by more than 20 collectors using 25 different types of smartphones, and the total number of AP is more than 500. Table 1 lists the detailed data information for each building, which tells the dataset diversity and complexity are high enough. The fingerprint points' distribution for each building is visualized in Fig. 5. And Table 2 gives the number of MAC repetitions between each building.

Experiments setting

The fingerprint data used for the classifier training is from one building, but the validation data may come from different buildings to test the adaptability of different classifiers. Table 3 shows the setting of the experiment with M representing the Model and T the Test.

Table 3 The different experimental test groups

Training data source	Testing data source	Experiment id
Building 0	Building 0	M0-T0
	Building 1	M0-T1
	Building 2	M0-T2
Building 1	Building 0	M1-T0
	Building 1	M1-T1
	Building 2	M1-T2
Building 2	Building 0	M2-T0
	Building 1	M2-T1
	Building 2	M2-T2

For example, M0-T1 means the data for model training come from Building0 and the validation data come from Building1.

Experiment results

To validate the effectiveness of the proposed method, the results are compared with the results with other methods in the relevant articles. Many comparative experiments are conducted. We test the performance of the Nearest Neighbor (NN), DT, and Ensemble Learning (EL) algorithms, including the bagging algorithms (Bagging and RF) and the boosting algorithms (XGBoost and GBDT). To test the feasibility of the proposed method

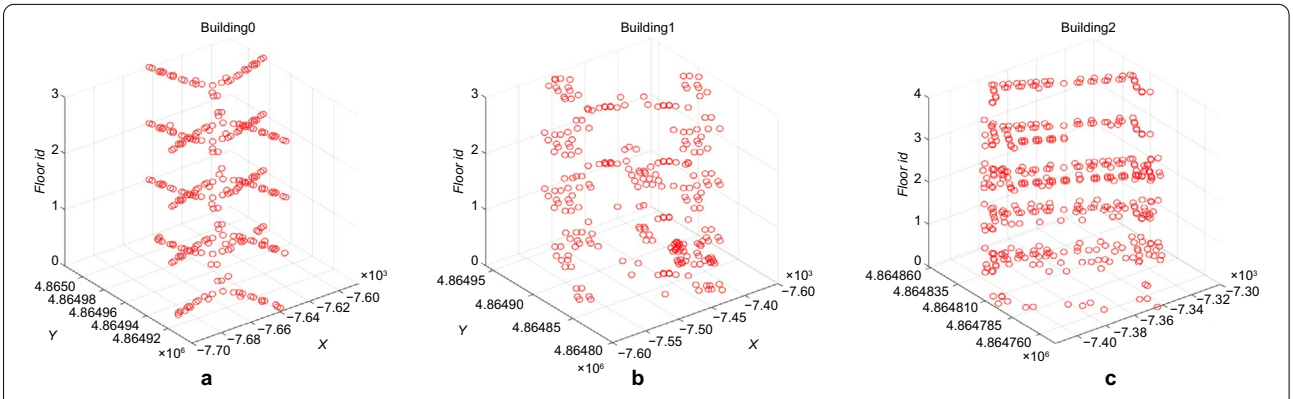

Fig. 5 The fingerprint points' distribution of each building

Table 4 The results of other articles and the method we proposed

Article	Mean positioning error	Floor judgment accuracy
Torres-Sospedra et al. (2014)	7.90 m	89.92%
Berkvens et al. (2016)	9.20 m	90.10%
Torres-Sospedra et al. (2015)	6.86 m	94.78%
Song et al. (2019)	11.78 m	96.03%
Nowicki and Wietrzykowski (2017)	–	T:92% (V:99%)
Proposed method 1 (XGBoost)	4.02 m	99.22%
Proposed method 2 (GBDT)	3.46 m	98.54%

*T represents the testing dataset, V is the validation dataset

for different buildings, the classifiers and the test samples from three different buildings are used for validation. In addition, some experiments are performed to test the effectiveness of the features, including the cases of 23 features, 57 features, and the low-scored features deleted.

Compared with existing methods

Since the UJIndoorLoc Dataset is widely used to test the performance of the fingerprint positioning algorithms, the result of the proposed method is compared with the results of other approaches listed in Table 4. We can see from the table that the proposed novel method has a better performance not only in the positioning error but also in the building and floor judgment accuracy. The first proposed method uses the XGBoost and achieves the mean positioning error of 3.42 m and the floor judgment accuracy of up to 99.40%. The second proposed method uses the GBDT and achieves the mean positioning error of 2.45 m and the floor judgment of 99.14%.

Comparison with popular algorithms

There are three main evaluation indices: floor detection accuracy, point matching accuracy, and mean positioning error. The point matching accuracy represents the rate of the final result matches the chosen testing fingerprint point.

The NN is one of the most popular methods in fingerprint positioning. We transfer the multi-classification into the binary classification in this paper, and the previous works show that tree models perform well in the binary classification. Thus, the DT and its enhanced algorithm-EL, including bagging algorithm and boosting algorithm, are chosen to evaluate the performance of the proposed method. The bagging algorithm includes the bagging and the RF, while the boosting algorithm includes XGBoost and GBDT. The detailed results are shown in Table 5. It is obvious that the performance of the single tree is much poorer than the EL. And the performance of the bagging algorithm is poorer than the boosting algorithm. As shown in Table 5, the positioning accuracy is slightly higher when using the GBDT. However, the XGBoost performs better in floor detection. Figure 6 shows the CDFs of different algorithms.

Classifiers trained by different buildings

The main objective of our method is to improve the adaptability of the classifier, making the classifier trained by one building usable in another building. The test data from three buildings are used to test the validity of the proposed method. The success rate of the floor judgment and point judgment, and the mean positioning error are recorded in the following tables. The performance of the DT, bagging algorithm, and boosting algorithm are tested, respectively. It is obvious that the boosting algorithm performs better than other methods, and the proposed method performs well even if the data come from different buildings. Tables 6, 7, 8 show the results of each building, and Figs. 7, 8, 9 show the CDFs of each method.

Table 5 The performance of different methods when the training data and validation data come from the same building

Algorithm	Building0			Building1			Building2		
	Floor accuracy (%)	Point accuracy (%)	Mean error (m)	Floor accuracy (%)	Point accuracy (%)	Mean error (m)	Floor accuracy (%)	Point accuracy (%)	Mean error (m)
NN	92.35	71.54	4.063	97.41	82.68	3.278	79.49	63.23	13.098
DT	37.89	2.73	23.602	46.43	6.12%	24.317	33.25	1.29	39.034
RF	98.44	71.88	2.102	97.96	73.98	3.050	96.39	74.23	1.880
Bagging	95.70	61.33	3.611	95.41	63.78	4.403	95.62	63.40	3.488
XGBoost	99.22	66.41	2.414	98.98	65.31	4.269	100.00	62.11	3.563
GBDT	100.00	78.52	1.399	98.98	76.53	2.873	98.45	74.49	3.089

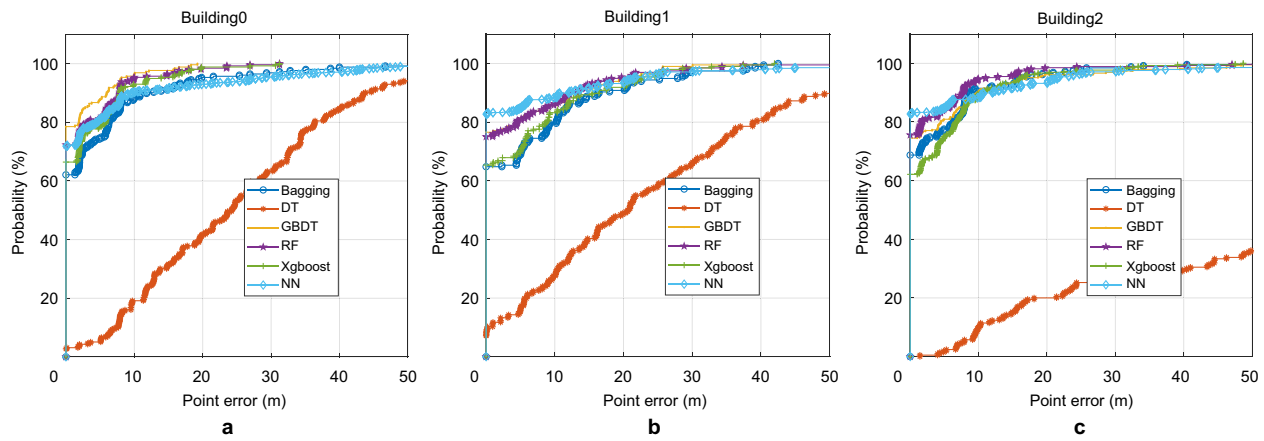

Fig. 6 The CDFs of different methods

Table 6 The results of different classifiers and testing samples using XGBoost

Model-test	GBDT			XGBoost		
	Floor success rate (%)	Point success rate (%)	Mean error (m)	Floor success rate (%)	Point success rate (%)	Mean error (m)
M0-T0	100.00	78.52	1.399	99.22	66.41	2.414
M0-T1	100.00	81.63	2.927	100.00	70.41	4.336
M0-T2	96.39	54.12	6.005	100.00	60.31	4.808
M1-T0	99.22	59.38	3.002	96.48	46.09	4.639
M1-T1	98.98	76.53	2.873	98.98	65.31	4.269
M1-T2	93.81	52.06	7.257	99.49	55.16	5.109
M2-T0	100.00	75.00	1.803	98.83	55.86	3.122
M2-T1	100.00	78.06	2.804	100.00	67.86	3.936
M2-T2	98.45	74.49	3.089	100.00	62.11	3.563
Mean	98.54	69.98	3.460	99.22	61.06	4.022

Table 7 The results of different classifiers and testing samples using Bagging

Model-test	Bagging			RF		
	Floor success rate (%)	Point success rate (%)	Mean error (m)	Floor success rate (%)	Point success rate (%)	Mean error (m)
M0-T0	95.70	61.33	3.611	98.44	71.88	2.102
M0-T1	96.43	63.78	6.821	98.98	71.94	5.455
M0-T2	87.63	41.50	8.949	96.65	56.44	6.408
M1-T0	94.92	51.17	3.872	98.44	64.45	3.033
M1-T1	95.41	63.78	4.403	97.96	73.98	3.050
M1-T2	95.62	39.43	7.253	99.23	64.69	3.493
M2-T0	95.70	48.44	4.277	100.00	69.14	2.102
M2-T1	96.94	66.33	5.057	99.49	70.41	4.720
M2-T2	95.62	63.40	3.488	96.39	74.23	1.880
Mean	94.89	55.46	5.303	98.40	68.57	3.583

Table 8 The results of different classifiers and testing samples using DT

Model-test	Floor success rate (%)	Point success rate (%)	Mean error (m)
M0-T0	37.89	2.73	23.602
M0-T1	52.55	6.12	29.683
M0-T2	32.22	1.29	48.224
M1-T0	37.50	2.34	21.025
M1-T1	46.43	6.12	24.317
M1-T2	35.05	1.03	42.775
M2-T0	39.84	3.13	22.828
M2-T1	51.53	7.65	19.929
M2-T2	33.25	1.29	39.034
Mean	40.70	3.52	30.157

Feature dimension reduction

23 features are chosen to test the proposed method. Table 9 shows the result by using the classifier trained by 23 features. To improve the dimension of the feature, we increase the number of the feature to 57, Table 6 shows its performance.

When we use XGBoost and GBDT, the score of each feature can be output after training. To improve the accuracy of each classifier, Figs. 10, 11, 12 show the score of each feature. But the low-score features are different for different classifiers. Thus, some common low-score features are deleted. The remaining features are used to train a new classifier, and the performance of the new classifier also is tested. Table 10 shows the performance after deleting some features. We can find no obvious improvement in the validity after deleting the low-score features. Figures 10, 11, 12 show the score of each feature in different buildings.

Conclusion and future work

To improve the performance of the indoor fingerprint-based positioning, it is a trend to use the method of ML or DL. However, current methods using the MAC-RSS pairs as the features face many problems, like low scene adaptability and the accuracy of the localization model. Thus, we proposed a novel method to solve these problems. To improve the model generalizable ability, we divided the samples into positive pairs and negative pairs and calculated the relative features rather than the absolute features from these pairs. Some methods were used to enhance the dimension of the features. Then the binary classification was used to replace the multi-classification, and the boosting algorithm was used to improve the accuracy of the classification model. The open-source dataset-UJIndoorLoc Dataset was used to test the performance of the proposed method. The results show that the proposed method performs better in floor judgment success rate and positioning error when compared with the NN and other binary classification models. Further studies are necessary, including the development of a method to construct more effective positive samples and negative samples and the employment of the DL rather than ML (Figs. 13, 14).

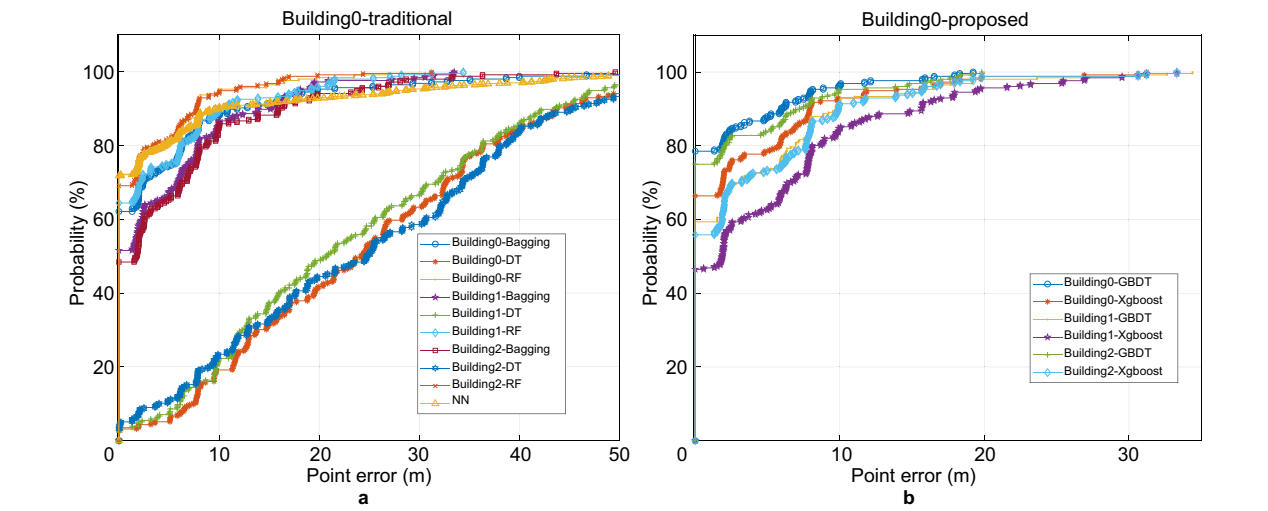

Fig. 7 The CDFs of Building0 using traditional methods and the proposed method

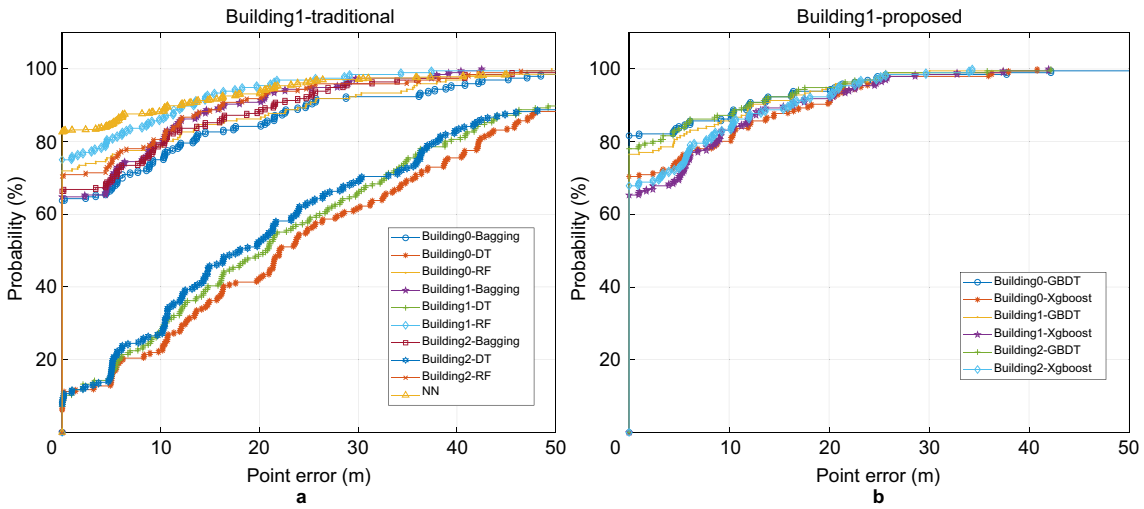

Fig. 8 The CDFs of Building1 using traditional methods and the proposed method

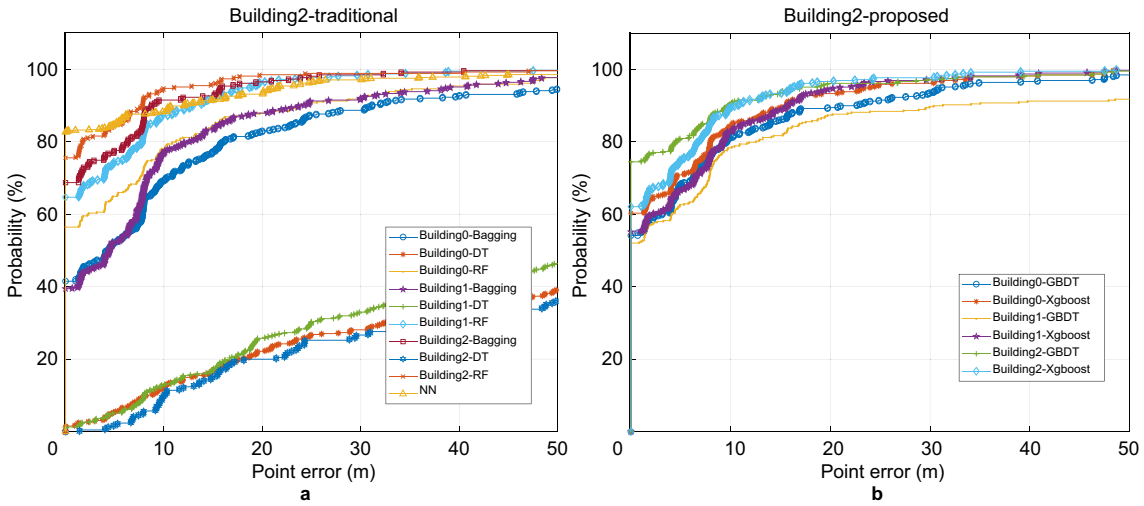

Fig. 9 The CDFs of Building2 using traditional methods and the proposed method

Table 9 The results of different testing samples using 23 features with XGBoost and GBDT

Model-test	XGBoost			GBDT		
	Floor success rate (%)	Point success rate (%)	Mean error (m)	Floor success rate (%)	Point success rate (%)	Mean error (m)
M0-T0	98.83	60.94	2.527	100.00	81.25	1.300
M0-T1	98.98	68.37	4.311	98.98	79.08	3.231
M0-T2	99.23	59.28	4.753	96.65	59.02	7.421
M1-T0	98.83	57.81	3.104	99.61	69.14	2.363
M1-T1	99.49	62.76	4.861	99.49	78.06	2.535
M1-T2	99.23	47.42	5.835	83.25	44.85	9.109
M2-T0	97.66	48.83	4.241	100.00	75.00	2.035
M2-T1	100.00	59.69	5.241	100.00	80.10	2.757
M2-T2	99.49	45.88	5.227	97.68	75.26	3.032
Mean	99.08	56.77	4.460	97.30	71.31	3.750

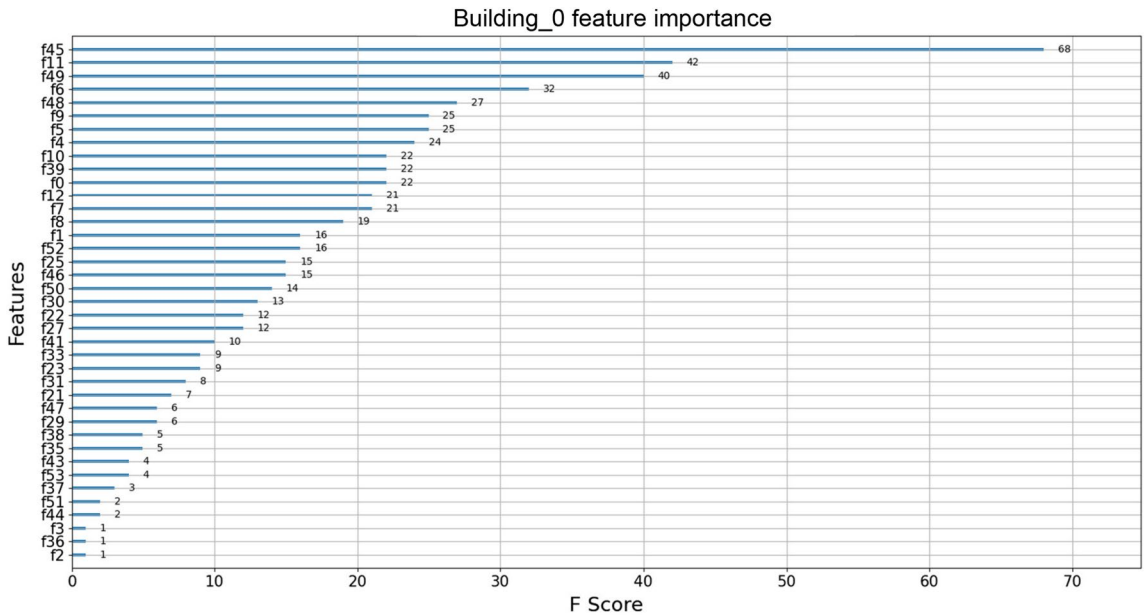

Fig. 10 The score of each feature using the data which comes from Building0

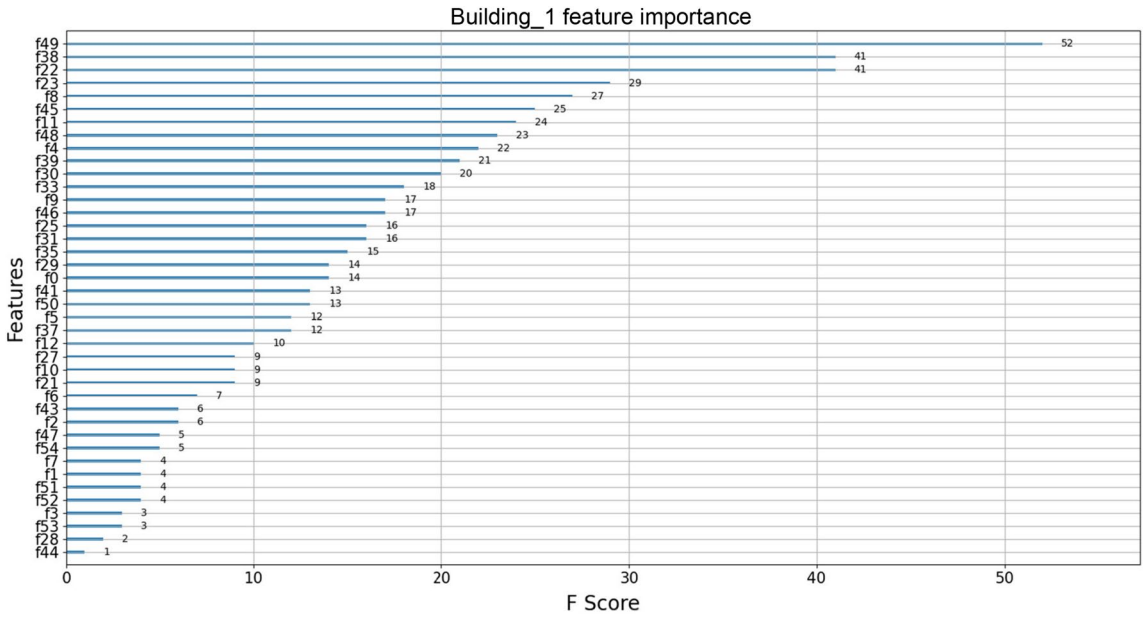

Fig. 11 The score of each feature using the data which comes from Building1

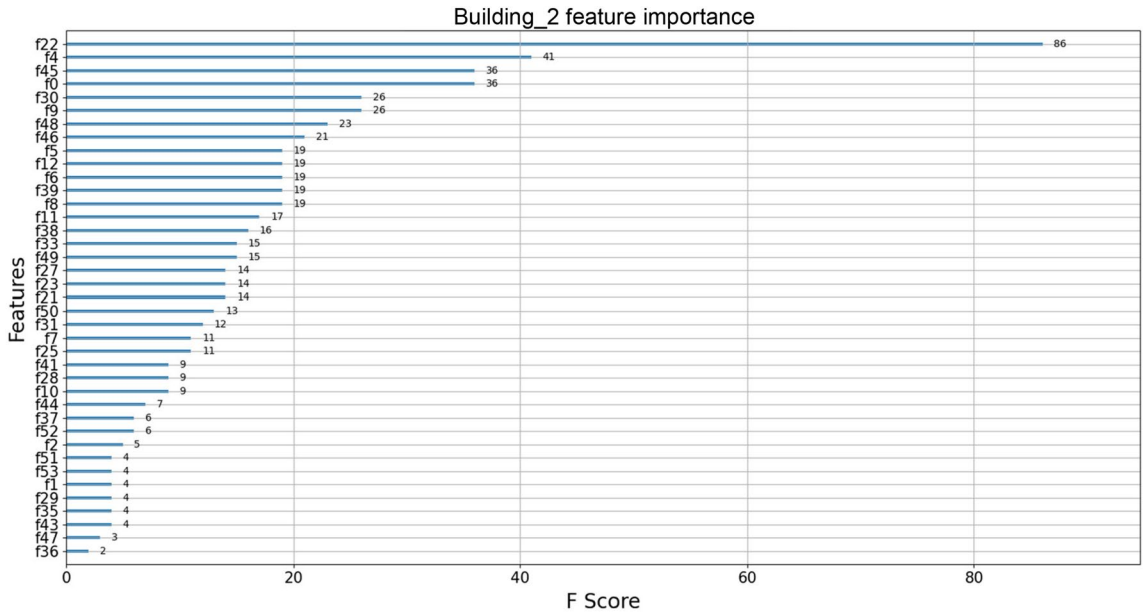

Fig. 12 The score of each feature using the data which comes from building2

Table 10 The results of different testing samples using XGBoost and GBDT after deleting the low-score Features

Model-test	GBDT			XGBoost		
	Floor success rate (%)	Point success rate (%)	Mean error (m)	Floor success rate (%)	Point success rate (%)	Mean error (m)
M0-T0	100.00	79.69	1.455	98.05	64.45	2.534
M0-T1	100.00	80.61	3.070	98.98	66.33	4.933
M0-T2	95.62	52.84	6.293	99.23	55.67	4.943
M1-T0	99.22	60.55	2.783	97.66	39.06	4.899
M1-T1	99.49	73.98	3.294	96.94	57.14	4.661
M1-T2	94.07	55.67	7.631	98.71	55.41	4.854
M2-T0	100.00	74.61	1.855	98.44	46.09	3.846
M2-T1	100.00	79.08	2.715	100.00	63.27	4.781
M2-T2	99.74	76.55	2.657	100.00	57.22	3.920
Mean	98.68	70.40	3.530	98.67	56.07	4.370

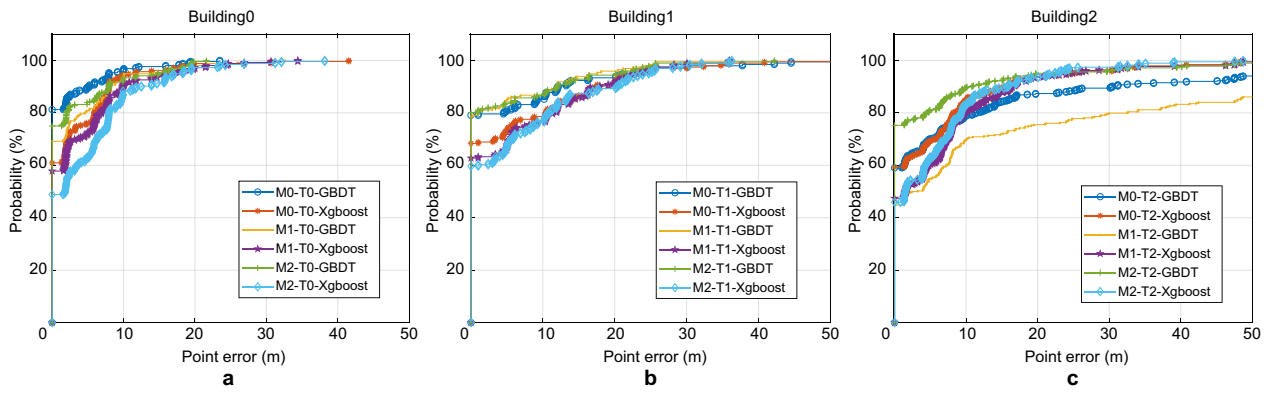

Fig. 13 The CDFs of the proposed method using partial features

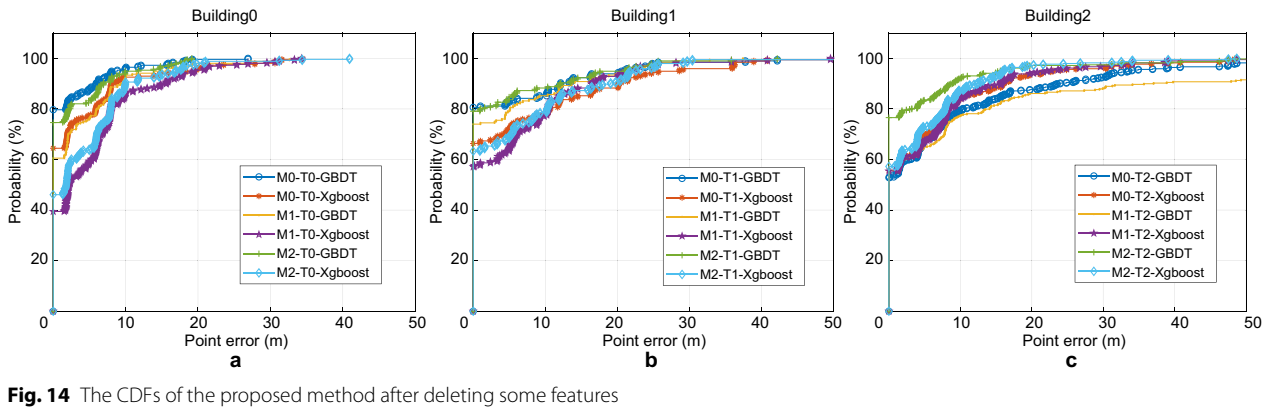

Fig. 14 The CDFs of the proposed method after deleting some features

Acknowledgements
Not applicable' for that section.

Authors' contributions
CXX proposed the idea, CXX carried out the programming and calculation; CXX wrote the draft, ZY, SX, YXS and WX edited and revised the manuscript. All authors read and approved the final manuscript.

Funding
Not applicable.

Availability of data and materials
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request, and the UJIndoorLoc dataset we used in this paper is open-source.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 21 July 2021 Accepted: 7 November 2021
Published online: 06 December 2021

References

Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), 2000* (Vol. 2, pp. 775–784). https://doi.org/10.1109/INFCOM.2000.832252

Berkvens, R., Weyn, M., & Peremans, H. (2016). Position error and entropy of probabilistic Wi-Fi fingerprinting in the UJIIndoorLoc dataset. *International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2016*, 1–6. https://doi.org/10.1109/IPIN.2016.7743691

Bi, J., Wang, Y., Li, X., Cao, H., Qi, H., & Wang, Y. (2018). A novel method of adaptive weighted K-nearest neighbor fingerprint indoor positioning considering user's orientation. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1177/1550147718785885

Chanama, L., & Wongwirat, O. (2018). A comparison of decision tree based techniques for indoor positioning system. In *2018 international conference on information networking (ICOIN)* (pp. 732–737). https://doi.org/10.1109/ICOIN.2018.8343215

Chen, G., Meng, X., Wang, Y., Zhang, Y., Tian, P., & Yang, H. (2015). Integrated WiFi/PDR/smartphone using an unscented Kalman filter algorithm for 3d indoor localization. *Sensors, 15*, 24595–24614. https://doi.org/10.3390/s150924595

Chen, L., Pei, L., Kuusniemi, H., et al. (2013). Bayesian fusion for indoor positioning using bluetooth fingerprints. *Wireless Personal Communications, 70*, 1735–1745. https://doi.org/10.1007/s11277-012-0777-1

Deng, Z., Fan, J., & Jiao, J. (2018). D-SVM fusion clustering algorithm based on indoor location. https://doi.org/10.1007/978-3-319-74521-3_27

El-Sheimy, N., & Li, Y. (2021). Indoor navigation: State of the art and future trends. *Satellite Navigation, 2*, 7. https://doi.org/10.1186/s43020-021-00041-3

El-Sheimy, N., & Youssef, A. (2020). Inertial sensors technologies for navigation applications: State of the art and future trends. *Satellite Navigation, 1*, 2. https://doi.org/10.1186/s43020-019-0001-5

Esmond, M., & Bernard, C. (2013). An improved neural network training algorithm for Wi-Fi Fingerprinting positioning. *International Journal of Geo-Information, 2*(3), 854–868. https://doi.org/10.3390/ijgi2030854

Feng, Y., Minghua, J., Jing, L., Xiao, Q., Ming, H., Tao, P., & Xinrong, H. (2014). Improved AdaBoost-based fingerprint algorithm for WiFi indoor localization. In *2014 IEEE 7th joint international information technology and artificial intelligence conference* (pp. 16–19). https://doi.org/10.1109/ITAIC.2014.7064997

Guo, S., & Pun, M. (2019). Indoor semantic-rich link-node model construction using crowdsourced trajectories from smartphones. *IEEE Sensors Journal, 19*(22), 10917–10934. https://doi.org/10.1109/JSEN.2019.2933746

Han, S., Zhao, C., Meng, W., & Li, C. (2015). Cosine similarity based fingerprinting algorithm in WLAN indoor positioning against device diversity. In *2015 IEEE international conference on communications (ICC)* (pp 2710–2714). https://doi.org/10.1109/ICC.2015.7248735

He, S., & Chan, S.-G. (2016). Wi-Fi fingerprint-based indoor positioning: recent advances and comparisons. *IEEE Communications Surveys & Tutorials, 18*(1), 466–490. https://doi.org/10.1109/COMST.2015.2464084

Hodes, T. D., Katz, R. H., Schreiber, E. S., & Rowe, L. (1997). Composable ad hoc mobile services for universal interaction. In *MobiCom'97 proceedings* (pp. 1–12). https://doi.org/10.1145/262116.262121.

Kaemarungsi, K., & Krishnamurthy, P. (2004). Modeling of indoor positioning systems based on location fingerprinting. In *IEEE INFOCOM 2004* (Vol. 2, pp. 1012–1022). https://doi.org/10.1109/INFCOM.2004.1356988

Kolakowski, M. (2020). Automatic radio map creation in a fingerprinting-based BLE/UWB localization system. *IET Microwaves, Antennas & Propagation, 14*(14), 1758–1765. https://doi.org/10.1049/iet-map.2019.0953

Lee, S., Kim, J., & Moon, N. (2019). Random forest and WiFi fingerprint-based indoor location recognition system using smart watch. *Human-Centric Computing and Information Sciences, 9*(1), 6. https://doi.org/10.1186/s13673-019-0168-7

Li, J., Gao, X., Hu, Z., et al. (2019). Indoor localization method based on regional division with IFCM. *Electronics, 8*(5), 559. https://doi.org/10.3390/electronics8050559

Li, Y., et al. (2021). Toward location-enabled IoT (LE-IoT): IoT positioning techniques, error sources, and error mitigation. *IEEE Internet of Things Journal, 8*(6), 4035–4062. https://doi.org/10.1109/JIOT.2020.3019199

Liu, J., Gao, K., Guo, W., et al. (2020). Role, path, and vision of "5G + BDS/GNSS." *Satellite Navigation, 1*, 23. https://doi.org/10.1186/s43020-020-00024-w

Liu, X.-Y., & Wang, X. (2020). Real-time indoor localization for smartphones using tensor-generative adversarial nets. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2020.3010724

Machaj, J., Brida, P., & Piché, R. (2011). Rank based fingerprinting algorithm for indoor positioning. In *2011 international conference on indoor positioning and indoor Navigation* (pp. 1–6). https://doi.org/10.1109/IPIN.2011.6071929

Naser, E.-S., & Li, Y. (2021). Indoor navigation: State of the art and future trends. *Satellite Navigation, 2*, 7. https://doi.org/10.1186/s43020-021-00041-3

Nowicki, M., & Wietrzykowski, J. (2017). Low-effort place recognition with WiFi fingerprints using deep learning. In R. Szewczyk, C. Zieliński, & M. Kaliczyńska (Eds.), *Automation 2017. ICA 2017. Advances in intelligent systems and computing.* (Vol. 550). Springer. https://doi.org/10.1007/978-3-319-54042-9_57

Shao, W., Luo, H., Zhao, F., Ma, Y., Zhao, Z., & Crivello, A. (2018). Indoor positioning based on fingerprint-image and deep learning. *IEEE Access, 6*, 74699–74712. https://doi.org/10.1109/ACCESS.2018.2884193

Song, C., & Wang, J. (2017). WLAN fingerprint indoor positioning strategy based on implicit crowdsourcing and semi-supervised learning. *ISPRS International Journal of Geo-Information, 6*, 356. https://doi.org/10.3390/ijgi6110356

Song, X., et al. (2019). A novel convolutional neural network based indoor localization framework with WiFi fingerprinting. *IEEE Access, 7*, 110698–110709. https://doi.org/10.1109/ACCESS.2019.2933921

Torres-Sospedra, J., Montoliu, R., Martínez-Usó, A., Avariento, J. P., Arnau, T. J., Benedito-Bordonau, M., & Huerta, J. (2014). UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In *2014 international conference on indoor positioning and indoor navigation (IPIN)* (pp. 261–270). https://doi.org/10.1109/IPIN.2014.7275492

Torres-Sospedra, J., Montoliu, R., Trilles, S., Belmonte, Ó., & Huerta, J. (2015). Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Systems with Applications, 42*(23), 9263–9278. https://doi.org/10.1016/j.eswa.2015.08.013

Yang, Z., Wu, C., Zhou, Z., Zhang, X., Wang, X., & Liu, Y. (2015). Mobility increases localizability: A survey on wireless indoor localization using inertial sensors. *ACM Computing Surveys, 47*(3), 1–34. https://doi.org/10.1145/2676430

Zhang, L., Liu, X., Song, J., Gurrin, C., & Zhu, Z. (2013). A comprehensive study of bluetooth fingerprinting-based algorithms for localization. In *2013 27th international conference on advanced information networking and applications workshops, 2013* (pp. 300–305). https://doi.org/10.1109/WAINA.2013.205

Zhuang, Y., et al. (2018). A survey of positioning systems using visible LED lights. *IEEE Communications Surveys & Tutorials, 20*(3), 1963–1988. https://doi.org/10.1109/COMST.2018.2806558

Zhuang, Y., Syed, Z., Li, Y., & El-Sheimy, N. (2016). Evaluation of two WiFi positioning systems based on autonomous crowdsourcing of handheld devices for indoor navigation. *IEEE Transactions on Mobile Computing, 15*(8), 1982–1995. https://doi.org/10.1109/TMC.2015.2451641

Zou, H., et al. (2020). Adversarial learning-enabled automatic WiFi indoor radio map construction and adaptation with mobile robot. *IEEE Internet of Things Journal, 7*(8), 6946–6954. https://doi.org/10.1109/JIOT.2020.2979413

Publisher's Note