 doi:10.20944/preprints202103.0215.v1

Article

JLGBMLoc: A Novel High-precision Indoor Localization Method Based on LightGBM

Lu Yin [1],Pengcheng Ma [2], and Zhongliang Deng [†]

[1] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China; inlu_mail@bupt.edu.cn
[2] School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China; mpc@bupt.edu.cn
† School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China.

Abstract: Wi-Fi based localization has become one of the most practical methods for mobile users in location-based services. However, due to the interference of multipath and high-dimensional sparseness of fingerprint data, the localization system based on received signal strength (RSS) is hard to obtain high accuracy. In this paper, we propose a novel indoor positioning method, named JLGBMLoc (Joint denoising auto-encoder with LightGBM Localization). Firstly, because the noise and outliers may influence the dimensionality reduction on high-dimensional sparseness fingerprint data, we propose a novel feature extraction algorithm, named joint denoising auto-encoder (JDAE), which reconstructs the sparseness fingerprint data for a better feature representation and restores the fingerprint data. Then, the LightGBM is introduced to the Wi-Fi localization by scattering the processed fingerprint data to histogram, and dividing the decision tree under leaf-wise algorithm with depth limitation. At last, we evaluated the proposed JLGBMLoc on UJIIndoorLoc dataset and Tampere dataset, experimental results show that the proposed model increases the positioning accuracy dramatically comparing with other existing methods.

Keywords: Indoor Localization, Wi-Fi Fingerprint, Denoising Auto-encoder, JLGBMLoc.

1. Introduction

In recent years, location based service (LBS) has developed rapidly. However, due to severe signal attenuation and multipath effects, general outdoor positioning facilities (such as GPS) can't work effectively in buildings [1]. Therefore, several types of indoor positioning technologies were proposed, such as wireless local area network (WLAN), visible light, cellular networks and their combination technologies [2,3]. The indoor positioning based on Wi-Fi signals has the advantages with convenient deployment, low hardware cost and high real-time performance. However, Wi-Fi based indoor positioning faces the problem of the volatility of Wi-Fi signals and high-dimensional sparseness of fingerprint [4]. This study focused on improving indoor positioning using Wi-Fi fingerprint.

Generally, a Wi-Fi system consists of some fixed access points (APs) [12]. Mobile devices (such as laptops and mobile phones) that connect to Wi-Fi can communicate directly or indirectly with each other. Received signal strength (RSS) of the AP is usually used to pre-built fingerprint database to infer the location of the mobile user. There are two stages in fingerprints positioning: offline stage and online stage [5]. The offline stage is to measure RSS readings of known locations from the surrounding access points and correlate them with these physical locations to build a fingerprint database. The collected data is training set; In the online phase, the real-time sampled RSS vector of the target is compared with stored fingerprints for positioning, where the location of the best matched fingerprint is selected as the target location, positioning result will be sent back to the requester.

Literature research faces two of key problems in fingerprint based localization. Firstly, the observed RSS vectors contain a large number of missing values due to the obstruction

 doi:10.20944/preprints202103.0215.v1

of out-of-range APs, random noise, signal fluctuation or scanning duration [6], especially inside large buildings, such as shopping malls and hospitals, which results in extremely data sparsity. Traditional data dimensionality reduction methods, including principal component analysis (PCA) [7] and linear discriminant analysis (LDA) [8], which treat all samples as a whole to find an optimal linear mapping projection with the smallest mean square error. But it has poor reduction effect on complex data. With the development of neural network, feature extraction and fusion have become more popular.

Another challenge is how to achieve high-precision and high-efficiency localization under multipath and noise fluctuations. The indoor propagation of Wi-Fi signals is easily affected by the human body, some obstacles, and walls, which affects the accuracy of fingerprint positioning. Traditional machine learning methods, including k-nearest neighbor (KNN) [9] and support vector machine(SVM) [10], are not effective in dealing with non-linear problems. Compared with these algorithms, the artificial neural network (ANN) [11] estimates the non-linear position from the input through the selected activation function and adjustable weights, and has the ability to approximate high-dimensional and highly nonlinear models. Notice that ANN is fully connected, the depth of neural network is directly related to the complexity of its calculation, which may directly affect the accuracy of positioning results. In [13], a deep neural network (DNN) for receiver autonomous integrity monitoring (RAIM), named RAIM-NET, is proposed to improve the performance of integrity monitoring, with a problem specifically designed for loss function. Hoang proposed a recurrent neural network for accurate received signal strength indicator (RSSI) indoor positioning [14], using the results of different types of RNN, including LSTM [15] and gated recurrent unit (GRU) [16]. However, these algorithms still face challenges such as spatial ambiguity and RSS instability. In [26], a convolutional neural network (CNN) based indoor localization system with WiFi fingerprints is proposed, which has a 95% accuracy of floor-level localization on UJIIndoorLoc dataset.

In summary, RSS based indoor positioning still faces problem that noise and outliers affect high-dimensional sparse fingerprint data, and it is difficult to achieve high accuracy and high efficiency under multipath and noise fluctuations. To solve the above problem, this paper focuses on a novel feature extraction algorithm to reconstruct sparse fingerprint data to obtain better feature representation. Moreover, in order to reduce the space complexity and low training speed due to the pre-sorting algorithm of the existing gradient boosting model, a novel positioning model is introduced to disperse the processed fingerprint data into histograms, and divide the decision tree under the leaf-wise algorithm with depth limitation, which solves the problem of large space occupation and improves the calculation speed. The main contributions of this work are summarized as follows:

1) Aiming at the problem of extracting key features from sparse RSS data and reducing the influence noise and outliers of dataset, we propose a novel feature extraction algorithm, named joint denoising auto-encoder (JDAE), which reconstructs the sparseness fingerprint data for a better feature representation and restores the fingerprint data.

2) To achieve higher positioning accuracy under high efficiency, the LightGBM is introduced to the Wi-Fi localization by scattering the processed fingerprint data to histogram, and dividing the decision tree under leaf-wise algorithm with depth limitation.

3) The proposed model is evaluated by UJIIndoorLoc [18] dataset and Tampere [19] dataset. Experimental results show the proposed model is superior to traditional machine learning methods, the room-level positioning accuracy can reach 96.05% on UJIIndoorLoc, which is nearly 10% higher than the DNN method [17], and the floor-level positioning accuracy can reach 95.45% on Tempere, which is more predominant than current advanced methods.

The rest of this article is organized as follows: Chapter II introduces the background. Chapter III describes the architecture and the process of positioning based on our proposed model. In Chapter IV, we describe the preprocessing datasets, optimize the parameters of the model through experimental research and compare it with several benchmarks of positioning accuracy. Finally, we summarize the contribution of this work in Chapter V.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

2. preliminary

2.1. Denoising Auto-encoder

Auto-encoder is an unsupervised algorithm that automatically learn features from unlabeled data, which give a better feature description than the original data [20], and complete automatic selection of features, as shown in Figure 1. Considering the datasets based on large buildings have strong sparsity, the output location information only depends on a small part of the dimensions of the input vector, which means auto-encoder can effectively reduce the dimension of data, and the necessary feature information get retained. This conclusion will be confirmed in subsequent experiments.

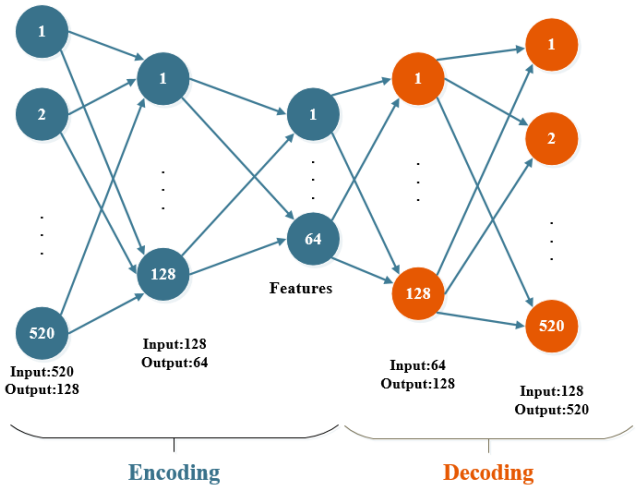


Figure 1. Auto-encoder Structure Chart

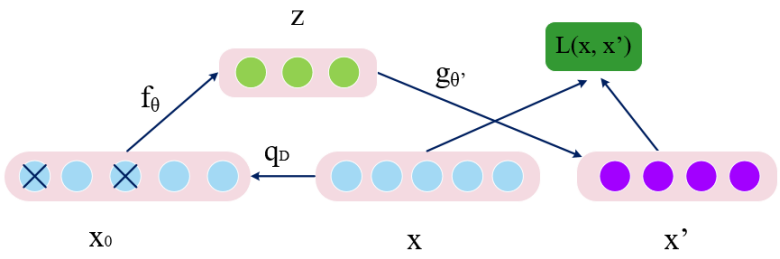


Figure 2. Denoising Auto-encoder Structure Chart

Unlike the auto-encoder, part of the input data is "corrupted" during the training process of denoising auto-encoder (DAE) [21]. The core idea of DAE is to encode and decode the "corrupted" original data, and then recover it. The real raw data can reduce the noise of the data and improve the robustness of the model. The principle is shown in Figure 2, where f_θ is the encoder, $g_{\theta'}$ is the decoder, and $L(x, x')$ is the loss function of the DAE network. The input data x is "corrupted" by noise according to the q_D distribution. The current problem is to adjust the network parameters, by calculating $L(x, x')$, to make sure the final output x' close to the original input x. In (1), W is the link weight from the input layer to the intermediate layer, and b is the bias of the intermediate layer. The signal x_0 is decoded by the decoding layer and transport to the output layer becomes z. In (2), W' is the link weight from the intermediate layer to the output layer, and b' is the bias of the output layer. x' is regarded as the prediction of x. We firstly randomly set the values of some dimensions to 0 according to a ratio to obtain a damaged vector x_0. Then the damaged vector is reconstructed to a lossless output x' as the input of the training model.

$$z = f_\theta(Wx_0 + b) \tag{1}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

$$x' = g_{\theta'}(W'z + b') \tag{2}$$

$$L(x, x') = ||x - x'||^2 \tag{3}$$

2.2. classification and regression tree

GBDT (Gradient Boosting Decision Tree) is an integrated learning of additive models based on regression trees [22]. The main idea is to continuously add weak learning functions, and perform feature splitting to grow a tree. GBDT uses classification and regression tree (CART) as a weak learning function, which refers to a decision tree that uses a binary tree as a logical structure to complete linear regression tasks. The CART classification tree algorithm uses Gini coefficient instead of information gain. The smaller Gini coefficient, the better model features. Assuming that the dataset has K categories, the probability of the k-th category is p_k, the Gini coefficient expression of the probability distribution is:

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{4}$$

For the sample set D, assuming that the sample has K categories, the number of k-th categories is C_k, then the Gini coefficient expression of sample D is:

$$Gini(D) = 1 - \sum_{k=1}^{K} (\frac{|Ck|}{|D|})^2 \tag{5}$$

According to value a of a certain feature A, divides D into D_1 and D_2, then under the condition of feature A, the Gini coefficient expression of sample D is:

$$Gini(D, A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \tag{6}$$

Therefore, for the CART classification tree, after the calculated Gini coefficient of each feature to the data set D, the feature A with the smallest Gini coefficient and the corresponding eigenvalue a are selected.

According to this optimal feature and optimal eigenvalue, the data set is divided into two parts D_1 and D_2, the left and right nodes of the current node are established at the same time. Until the Gini coefficient is less than the threshold, the decision tree subtree is returned, and the current node stops recursion. Notice that each time a tree is added, actually is to learn a new basic learner $h(.)$ to fit the final prediction.

3. system design

3.1. system model

The LightGBM used in this paper is an improvement based on the algorithm of GBDT. Assume that the region of interest has N APs and M reference points (RPs), the RSS input set can be defined as $f = \{f_1, f_2, ..., f_M\}$, and the corresponding location set is $l = \{l_1, l_2, ..., l_M\}$. The GBDT algorithm can be regarded as an additive model composed of K trees:

$$g(f_i) = \sum_{k=1}^{K} h_k(f_i), i \in 1, 2, ..., M \tag{7}$$

where $g(f_i)$ represents the predicted output, which is the predicted position in the model, $f_i = \{F_{i1}, F_{i2}, ..., F_{iN}\}$ is the RSS value set of the i-th sample, and F_{ij} is the j-th eigenvalues (i.e.RSS value) of RP_i. Obviously, our goal is to make the predicted value $g(f_i)$ of the tree group as close as possible to the true value $l_i = (x_i, y_i)$, and have the largest possible generalization ability. According to the characteristics of the sample, each tree will fall into the corresponding leaf node and correspond to a score. After completing training and getting K trees, the score corresponding to each tree is added to get the predicted value of the sample. In each iteration, on the basis of the existing tree, a tree is added to

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

fit the residual between the prediction result of the previous tree and the true value. The integrated learner obtained by $(t-1)$-th iteration is $g_{t-1}(f)$, the focus of the t-th training is to minimize the loss function (8) with the square loss functions (9) and (10):

$$L(l, g_t(f)) = L(l, g_{t-1}(f) + h_t(f)) \tag{8}$$

$$L(l, g_t - 1(f) + h_t(f)) = (l - g_{t-1}(f) - h_t(f))^2 = (r - h_t(f))^2 \tag{9}$$

$$r = l - g_{t-1}(f) \tag{10}$$

where r represents the residual. Each step of the GBDT algorithm needs to fit the residual of the previous model when generating the decision tree, and use the fastest descent approximation method, which means the negative gradient of the loss function is used as the approximate value of the residual in the lifting tree algorithm. The negative gradient of the loss function of the i-th sample in the t-th iteration is:

$$r_{it} = -[\frac{\partial \mathrm{L}(l_i, g(f_i))}{\partial g(f_i)}]_{g(f)=g_{t-1}(f)} \tag{11}$$

the residual obtained in the previous step is used as the new true value of the sample, and the data (f_i, r_{it}), $i = 1, 2, .., N$ is used as the next tree training data to obtain a new regression tree, the corresponding leaf node area is R_{jt}, $j = 1, 2, ..., J$, where J is the number of leaf nodes of the t-th regression tree. For leaf area j, we calculate the best fit value as:

$$\gamma_{jt} = \arg\min_{\gamma} \sum_{f_i \in R_{jt}} L(l_i, g_{t-1}(f_i) + \gamma) \tag{12}$$

Then, we update the strong learner and get the final learning function $g_K(f)$.The gradient boosting algorithm improve the robustness of data outliers through the loss function, which is greatly improved compared to the traditional machine learning algorithm.

$$g_t(f) = g_{t-1}(f) + \sum_{j=1}^{J} \gamma_{jt} I(f \in R_{jt}) \tag{13}$$

GBDT can handle various types of data flexibly. However, due to the dependence between weak learners, it is difficult to train data in parallel, which results in relatively low operating efficiency of the model. Therefore, high dimensional data will increase the complexity of the model. LightGBM is a high-performance gradient boosting framework based on decision tree algorithm released by Microsoft in 2017 [23], which can be used in sorting, classification, regression and other machine learning tasks. LightGBM has been optimized on the GBDT algorithm to speed up the training of GBDT model without compromising accuracy.

In the improved gradient boosting model based on Wi-Fi positioning, XGBoost [24] uses the pre-sorting algorithm to reduce the amount of calculation to find the best split point. But it still needs to traverse the positioning data set during the node splitting process, which increases the space complexity and training speed. Compared with XGBoost, LightGBM uses the histogram algorithm to process the positioning data set and the leaf-wise split strategy in the process of Wi-Fi-based positioning, which solves the problem of large space consumption due to pre-sorting and improves the calculation speed.

Firstly, in our positioning fingerprint, AP_j is regarded as the j-th feature of fingerprint data, and $F_j = \{F_{1j}, F_{2j}, ..., F_{Mj}\}$ is defined as a set of eigenvalues contained in AP_j. Then, a histogram decision tree algorithm is imported to discretize F_j into a histogram with a width of k. Instead of the traditional pre-sorting idea, each of these precise and continuous values is divided into a series of discrete domains. The histogram accumulates the required statistics according to the discrete value, and traverses to find the best positioning AP feature and the corresponding eigenvalue as the segmentation point. No additional storage of pre-classification results is needed, only discrete values of features can be saved, and

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

memory consumption can be reduced to one-eighth of the original value. The histogram is shown in Figure 3.

Considering the high-dimensional sparsity of the fingerprint data, the features represented by many APs are mutually exclusive, which means they usually don't take non-zero values at the same time. According to the exclusive feature bundle (EFB) algorithm of LightGBM, the complexity of fingerprint feature histogram construction changes from $O(data * feature)$ to $O(data * bundle)$, and $bundle << feature$, which greatly accelerates the training speed of gradient boost model without affecting the positioning accuracy.

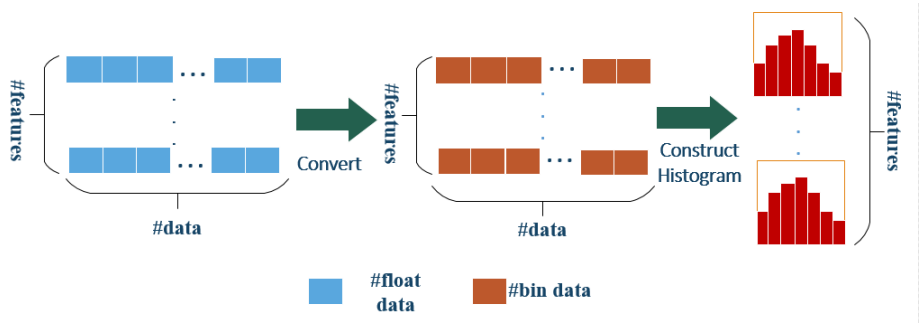


Figure 3. Histogram algorithm of LightGBM

Secondly, the traditional decision tree splitting strategy is to use level-wise to find the best positioning AP feature and the corresponding feature value as the split point. However, the AP feature of the same layer are treated indiscriminately, and many APs have lower split gain, which brings unnecessary cost. Therefore, a leaf-wise algorithm with depth limitation is used to find the feature with the largest split gain, which means the best feature of the fingerprint data can be found from all current leaves, and then split, to reduce more errors and obtain better accuracy. As shown in Figure 4, compared with level-wise, leaf-wise reduce more errors and get better accuracy when the number of splits is the same; However, when leaf-wise grow deeper, which causes decision tree over-fitting. Therefore, the maximum depth limit is added to prevent over-installation and ensure high efficiency.

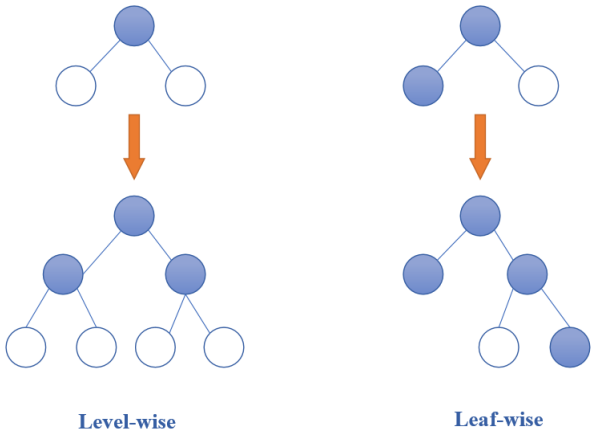


Figure 4. The generation strategy of tree

3.2. Feature Extraction Algorithm

Since each iteration of the gradient boosting algorithm adjusts the sample according to the prediction result of the previous iteration, as the iteration continues, the bias of the model will continue to decrease, which lead to model more sensitive to noise. In indoor positioning dataset, the outliers caused by multipath signals and NLOS will have an impact on training of the database.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

To solve the problem of extracting key features from sparse RSS data and reducing the influence noise and outliers of dataset, we propose a feature extraction algorithm, called joint denoising auto-encoder (JDAE), aiming at extracting key features from sparse RSS data and reducing the influence of noise and data outliers. The architecture of JDAE is shown in Figure 5. From input layer to feature layer is the part of auto-encoder, x means the input RSS data, $h^{(1)}$ is the hidden layer of auto-encoder, and f means the feature data processed by auto-encoder. This part is to extract key features from sparse Wi-Fi data. After

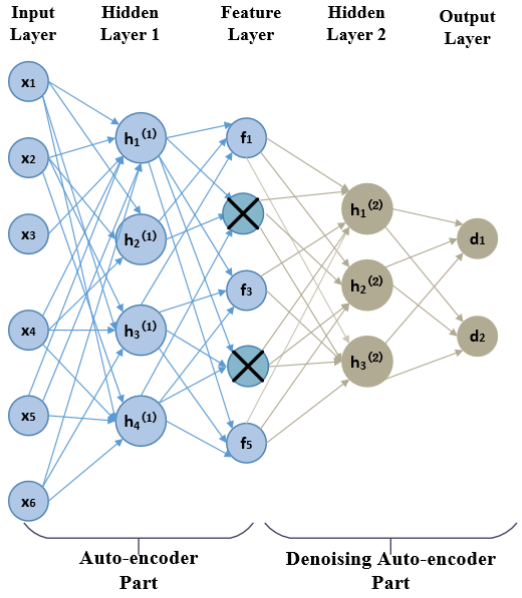


Figure 5. The architecture of JDAE

getting reconstructed features, next part is to introduce denoising auto-encoder to reduce the influence of noise and data outliers. The denoising auto-encoder randomly partially use the damaged input to solve the identity function risk, to make the auto-encoder denoised. The dropout in the auto-encoder network refers to randomly letting the weights of some hidden layer nodes of the network not work during model training. We apply dropout layyer to the input layer instead of the hidden layer. The damage ratio generally does not exceed 0.5, and Gaussian noise can also be used to damage the data. The feature is robustly obtained from the damaged input, which can be used to restore its corresponding noise-free fingerprint data. The "⊗" part in the feature layer means "corrupted" features according to our setting, and $h^{(2)}$ is the hidden layer of denoising auto-encoder. After processing dataset by JDAE, the output layer d is imported to the LightGBM model.

3.3. System Architecture

The positioning method based on LightGBM is divided into two stages, the offline training stage and online positioning stage. In training stage, RSS of each predefined RP are collected in the database, and RP_i has a corresponding fingerprint vector f_i={F_{i1}, F_{i2}, ..., F_{iN}} at its location $l_i(x_i, y_i)$, where N is the number of available features (i.e.RSS) of all APs. Note that x_i and y_i represent position coordinates, which are different from the meaning in the feature extraction diagram above. After standardizing the dataset, the proposed JDAE is introduced, aiming at extracting key features from sparse RSS data and reducing the influence of noise and data outliers. And then, LightGBM is imported to classify the processed data, and adjust the input parameters according to the results to obtain the optimal model; In online stage, the proposed model will localize each location by matching received fingerprinting measurement and send back the localization position to the mobile user. The detailed algorithm is shown in Figure 6. The mapping relationship between location and Wi-Fi signal data is learned through LightGBM.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

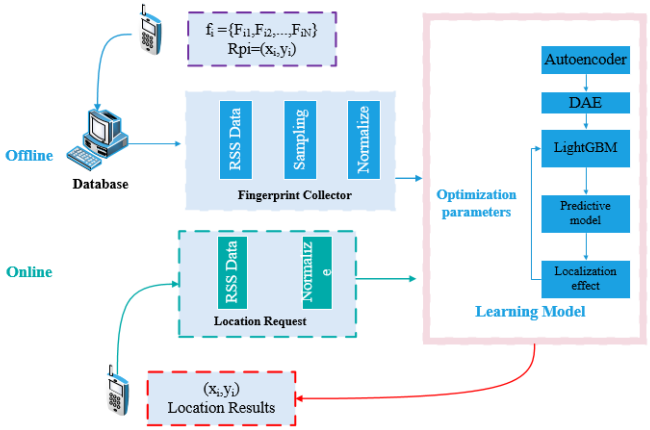


Figure 6. System Architecture

The idea of training the processed dataset is to transform the positioning problem into a multi-classification problem through position discretization, and each position corresponds to a category. Then, the samples are trained and the results of each decision tree are fused to get the final classification result. The steps of using the algorithm to train the fingerprint are as follows:

1) Firstly, a certain location is selected as the sampling point, the Wi-Fi fingerprint data is collected as all the characteristics of the sample. The histogram method algorithm is used to discretize the eigenvalues of the sample into K integers, and construct a histogram of width K for each feature. Then, according to the discrete value of the histogram, each AP point is used as the feature of the dataset, and the AP point corresponding to the minimum loss function value and the corresponding eigenvalue is calculated as the best split point for each iteration;

2) In order to prevent the built fingerprint database model from being too complicated and over-fitting, it is necessary to limit each split of the node. Only when the gain is greater than the threshold, the split is performed and when a tree reaches the maximum depth, it stops continuing to split;

3) When generating a decision tree, the gradient boost algorithm is used to make the predicted result continuously approach the real result, and offline training is completed through the learning of multiple decision trees. In online positioning stage, each testing Wi-Fi data is normalized, and sent to the trained multi-classification model for positioning.

4. Experiment evaluation

4.1. Data Preprocessing

The UJIIndoorLoc [18] dataset used in this paper covers three buildings of Jaume I University, with 4 floors and an area of nearly 110.000 square meters. It can be used for classification (for example, actual building and floor identification) or regression (estimation of longitude and latitude). It was created by more than 20 different users and 25 android devices. The database consists 19937 sets of training data and 1111 sets of testing data. As shown in Table 1, the 529 attributes contain RSS values, the coordinates of the locations and other useful information. Each Wi-Fi fingerprint can be characterized by the detected APs and corresponding RSS values. One Wi-Fi fingerprint consists of 520 intensity RSS values.

On UJIIndoorLoc dataset, the value of the input RSS data varies from -104 dbm to 0 dbm, and is nomalized for model training. In [26], different data representations of RSS fingerprints may affect the success rate and error. For any AP that is not detected in a measurement, its RSS value is marked as 100 dbm, and we denote these RSS values as 0.

$$\text{Normalized} = \begin{cases} 0, RSS_{ij} = 0 \\ (\frac{RSS_{ij} - \min}{-\min})^{\beta}, otherwise \end{cases} \quad (14)$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

Table 1: The Information of UJIIndoorLoc.

Attribute	Information
001	(AP001): Intensity value for AP001
...	...
520	(AP520): Intensity value for AP520
521	Longitude
522	Latitude
523	Floor ID
524	Space ID

Table 2: Parameter Settings.

Parameter	Setting
Subsample	0.8
Lose	MSE
Early stoping patience	5
Batch size	60
Feature fraction	0.8

where i is the AP identifier, RSS_{ij} the j-th RSS value of RP_i, the minimum value is the lowest RSS value considering all fingerprints in the database and the AP, and β is a mathematical constant. The results in [26] show that the normalized data tends to express the RSS value with the best performance, and tame the fluctuating RSS signal. Therefore, the normalized data is used to represent the Wi-Fi fingerprint in this paper.

Tampere dataset covers two buildings of Tampere [19] University of Technology. In the first building, there are 1478 sets of training data and 489 sets of testing data. 312 attributes include Wi-Fi fingerprints (309 AP) and coordinates. The intensity value is expressed as a negative integer ranging from -100dBm to 0dBm. The Wi-Fi fingerprint consists of 309 intensity values. In the second building, there are 583 sets of training data and 175 sets of testing data, including 357 attributes of Wi-Fi fingerprints (354 AP) and coordinates. The Tampere dataset uses floor height as floor representation instead of floor number. In this experiment, adjusted optimal model is verified with the Tampere to test the performance of JLGBMLoc.

The experiment is equipped with a laptop with Intel i5-6300 CPU, using python-3.7.6 on tensorflow environment to implement the model building. The parameters used in the initial optimization are shown in Table II.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2 \tag{15}$$

The loss function is the mean square error (MSE). Training batch is set to 60, and the patience parameter in the early stop is set to 5, feature fraction takes 0.8, which is equivalent to the learning rate. After one iteration, the weight of the leaf node will be multiplied by the coefficient. The purpose is to weaken the influence of each tree to make sure the later decision tree has more learning space.

4.2. Performance Evaluation of JDAE

The performance of the model is evaluated by comparing the performance of the model with the state-of-the-art methods. Two datasets, UJIIndoorLoc and Tampere are used for experimental research. The ratio of the number of correct matching positions to the total number of positions is used as the accuracy rate to evaluate the effects of each

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

method. The accuracy in this work is defined as follows, N_A means correctly predicted number of samples, and N means total number of samples.

$$Accuracy = \frac{N_A}{N} \tag{16}$$

Firstly, we use UJIIndoorLoc to optimize the parameters of the model, adjusted optimal model is verified with the Tampere dataset to test the performance of JLGBMLoc with different datasets. We firstly train the model in floor positioning, and then use the trained model to test room positioning accuracy.

The fixed parameters of LightGBM and default values is given in Table 2, and used alone to train UJIIndoorLoc without feature extraction. As shown in Figure 7, after completing the iteration, the loss function value reaches 0.51. The positioning accuracy of testing data finally reached 91.04%, which is nearly 10% higher than the DNN method in floor-level positioning [17]. The running time of LightGBM is 5.5 seconds, the speed of which is almost two times higher than XGBoost [24].

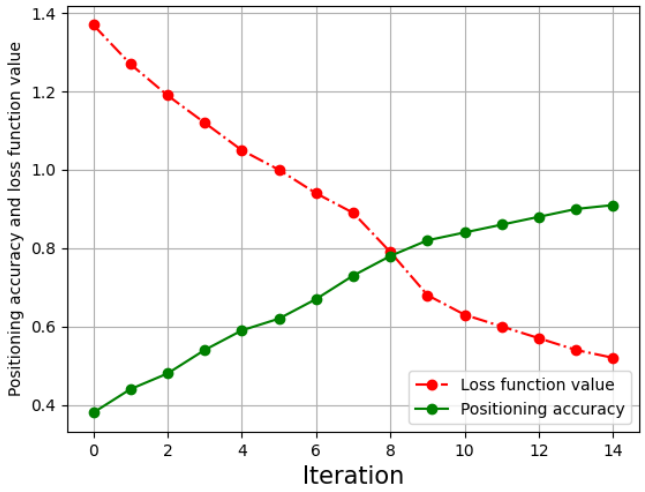
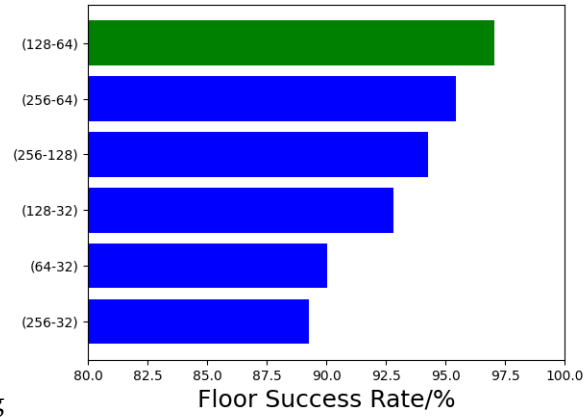


Figure 7. LightGBM localization without feature extraction

Then, different auto-encoder models are built to evaluate the best performance. The comparison result is shown in Figure 8. When the hidden layer and output layer are set to 128 and 64 respectively, the floor success accuracy rate reaches the highest 95.59%. Therefore, for auto-encoder, we choose the hidden and output layer to be 128 and 64.



Selection.png

Figure 8. Effect of different auto-encoder models on floor localization.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

The weights of the denoising auto-encoder is initialized. If the weight of the network is initialized too small, the signal will gradually shrink during transmission between layers and it will be difficult to produce any effect. The initialization method will automatically adjust the most appropriate random distribution according to the number of input and output nodes of a certain layer of network, which is to make the weight meet the 0 mean value. Assuming the number of input nodes in input dimension and the number of output nodes in output dimension, the variance of uniform distribution is $2/(input + output)$, and the form of random distribution can be uniform distribution. Here we make the input and output dimensions equal. The CDF of two methods are shown in Figure 9. Compared with single LightGBM model, the accuracy has been further improved. Not only that, our JDAE method is 8% more accurate than the method using a single auto-encoder in [25].

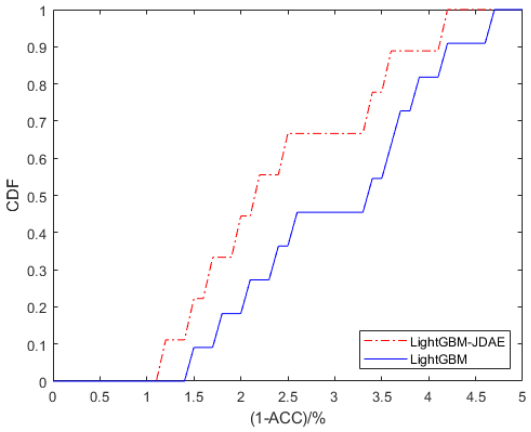


Figure 9. CDF comparison of the two methods

4.3. LightGBM Parameter Optimization

Firstly, we tune *num_leaves*, which is an important parameter to improve accuracy. Maximum depth represents the depth of the set tree, the greater the depth, the greater possibility of overfitting. Due to the leaf-wise algorithm used by LightGBM, number of leaves is used when adjusting the complexity of the tree. The approximate conversion relationship is:

$$\text{num\_leaves} = 2^{(\text{max\_depth})} - 1 \tag{17}$$

edefYour command was ignored. Type I <command> <return> to replace it with another comma

Figure 10. Optimization of learning rate and *num_leaves*

Secondly, for the regularization parameters, *lambda_l*1 and *lambda_l*2 effectively prevent the model from over-fitting. After adjusting our model, *lambda_l*1 and *lambda_l*2 is taken as 0.01, and the optimal accuracy rate in floor classification reaches 97.07%. And then, learning rate is adjusted. A small learning rate converges slowly, but can reduce the loss value to a lower value. When the learning rate is set too large, the gradient may oscillate around the minimum value, and may even fail to converge. The initial learning rate is 0.05. When the learning rate is 0.6, the accuracy reaches 99.32%, and when it gets 1, the accuracy drops to 95%. Therefore, The learning rate is set to 0.6. The result analysis is shown in Figure 10.

Finally, we optimize the parameter *min_split_gain*, which means the minimum gain for splitting the decision tree. After testing, the optimal parameter is 0.02. The parameters of the model is shown in Table III. We use the optimized model to position, randomly select about 1000 sets of data covering 50 rooms in the UJIIndoorLoc dataset for testing. We compare JLGBMLoc with DNN [17], CNNLoc [26] and LightGBM. The room success

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 March 2021 doi:10.20944/preprints202103.0215.v1

Table 3: Parameter Tuning.

Parameter	Value
learning rate	0.6
lambda_l1	0.01
lambda_l2	0.01
num_leaves	30
max_depth	5
min_split_gain	0.02

Success Rate.png

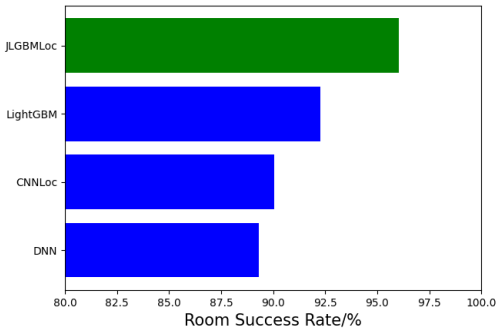


Figure 11. Room success rate comparison of different models.

rate comparison is shown in Figure 11, and the room accuracy rate reaches 96.05%. Due to the complexity of room-level positioning will be much higher than floor positioning, the accuracy will be slightly reduced. Experiments show our proposed model can achieve room-level positioning, and the accuracy is more predominant than current advanced methods.

4.4. Model Comparison

Compare.png

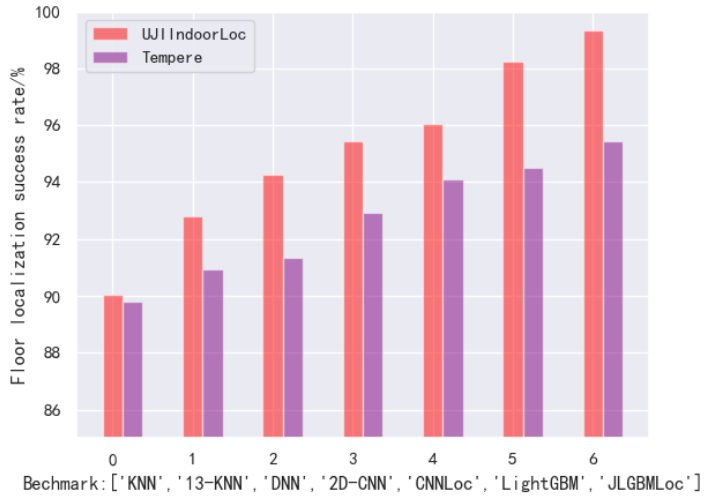


Figure 12. Localization result comparison on two datasets.

The performance of the model is evaluated by comparing JLGBMLoc with several state-of-the-art methods. We applied a unified extraction algorithm on UJIIndoorLoc, and obtained 2198 samples in the verification set and 1111 samples in the test set. We use the floor success rate to illustrate the comparison results. Not only that, we test the accuracy

 doi:10.20944/preprints202103.0215.v1

of position regression in Tempere. Benchmark methods include KNN [9], 13-KNN, DNN [17], CNN and CNNLoc [26]. Model comparison is shown in Fig. 12. The floor success rate of JLGBMLoc on UJIIndoorLoc is 99.32%, which on Tempere is 95.45%. The results show that the performance of our model is better than other benchmarks, which proves its high accuracy and scalability in different scenarios and datasets.

5. Conclusion

In this paper, we proposed a novel indoor positioning method, named JLGBMLoc. A novel feature extraction algorithm was proposed to reconstruct the sparseness fingerprint data, and LightGBM was introduced to the Wi-Fi localization. We evaluated the proposed JLGBMLoc on UJIIndoorLoc dataset and Tampere dataset, experimental results showed the proposed method has a room-level positioning accuracy of 96.05% on the UJIIndoorLoc, and floor-level accuracy of 95.49% on the Tampere. Experimental results proved the proposed JLGBMLoc increases the positioning accuracy dramatically comparing with other existing methods.

Author Contributions: conceptualization, Lu Yin and Pengcheng Ma; data curation, Lu Yin and Pengcheng Ma; methodology, Lu Yin and Pengcheng Ma; project administration, Zhongliang Deng; resources, Lu Yin and Pengcheng Ma; software, Pengcheng Ma; supervision, Lu Yin; validation, Lu Yin; writing—original draft, Pengcheng Ma; funding acquisition, Lu Yin; writing—review and editing, Pengcheng Ma; related work, Zhongliang Deng. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 61801041 and the Natural Science Foundation of Beijing Municipality under Grant L191003 and the State Key Laboratory of Information Photonics and Optical Communications (BUPT) under Grant IPOC2019ZT09, P. R. China.

Informed Consent Statement: Not applicable.

Data Availability Statement: The UJIIndoorLoc dataset and the Tempere dataset can be found here: https://archive.ics.uci.edu/ml/datasets.php. The UJIIndoorLoc dataset covers three buildings of Jaume I University, with 4 floors and an area of nearly 110.000 square meters.

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

References

1. Basiri A, Lohan E S, Moore T, et al. Indoor location based services challenges, requirements and usability of current solutions[J]. Computer Science Review, 2017, 24: 1-12.
2. Iannucci P A , Narula L , Humphreys T E . Cross-Modal Localization: Using automotive radar for absolute geolocation within a map produced with visible-light imagery[C]// 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2020.
3. Ren Jianxin, Zi Junlin, Song Kaiyue.Indoor Position Method Based on WLAN[J]. Navigation Positioning and Timing, 2020, 7(02): 110-116.(in Chinese).
4. Narupiyakul L , Sanghlao S , Yimwadsana B . An Indoor Navigation System for the Visually Impaired Based on RSS Lateration and RF Fingerprint[M]// Smart Homes and Health Telematics, Designing a Better Future: Urban Assisted Living. Springer, Cham, 2018.
5. Laoudias C, Moreira A, Kim S, et al. A survey of enabling technologies for network localization, tracking, and navigation[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3607-3644.
6. Apostolo, Guilherme Henrique , I. G. B. Sampaio , and José Viterbo. "Feature selection on database optimization for Wi-Fi fingerprint indoor positioning." Procedia Computer Science 159(2019):251-260.
7. Cadima, Jorge, Jolliffe, et al. Principal component analysis: a review and recent developments[J]. Philosophical Transactions of the Royal Society Mathematical Physical & Engineering Sciences, 2016.
8. Chu D, Liao L Z , Ng K P, et al. Incremental Linear Discriminant Analysis: A Fast Algorithm and Comparisons[J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 26(11):2716-2735.
9. Xie Y, Wang Y, Nallanathan A, et al. An improved K-nearest-neighbor indoor localization method based on spearman distance[J]. IEEE signal processing letters, 2016, 23(3): 351-355.
10. Bailey T L . MD-SVM: a novel SVM-based algorithm for the motif discovery of transcription factor binding sites[J]. Bioinformatics, 2019, 28(1):p.56-62.

11. Satu M S , Akter T , Uddin M J . Performance Analysis of Classifying Localization Sites of Protein using Data Mining Techniques and Artificial Neural Networks[C]// 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE). IEEE, 2017.
12. S. Han, Y. Li, W. Meng, C. Li, T. Liu and Y. Zhang, "Indoor Localization With a Single Wi-Fi Access Point Based on OFDM-MIMO," in IEEE Systems Journal, vol. 13, no. 1, pp. 964-972, March 2019, doi: 10.1109/JSYST.2018.2823358.
13. Sun, Y. RAIM-NET: A Deep Neural Network for Receiver Autonomous Integrity Monitoring. Remote Sens. 2020, 12, 1503.
14. Hoang M T, Yuen B, Dong X, et al. Recurrent neural networks for accurate RSSI indoor localization[J]. IEEE Internet of Things Journal, 2019, 6(6): 10639-10651.
15. Voigtlaender P, Doetsch P, Ney H. Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks[C]// 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE, 2017.
16. Tang Y , Huang Y , Wu Z , et al. Question detection from acoustic features using recurrent neural network with gated recurrent unit[C]// IEEE International Conference on Acoustics. IEEE, 2016.
17. Zhang W , Liu K , Zhang W , et al. Deep Neural Networks for wireless localization in indoor and outdoor environments[J]. Neurocomputing, 2016, 194(jun.19):279-287.
18. Torres-Sospedra J, Montoliu R, Martínez-Usó A, et al. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems[C]//2014 international conference on indoor positioning and indoor navigation (IPIN). IEEE, 2014: 261-270.
19. Lohan E S, Torres-Sospedra J, Leppäkoski H, et al. Wi-Fi crowdsourced fingerprinting dataset for indoor positioning[J]. Data, 2017, 2(4): 32.
20. Walker J , Doersch C , Gupta A , et al. An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders[C]// European Conference on Computer Vision. Springer International Publishing, 2016.
21. Grais E M , Plumbley M D . Single Channel Audio Source Separation using Convolutional Denoising Autoencoders[J]. 2017.
22. Wang W, Li T, Wang W, et al. Multiple Fingerprints-Based Indoor Localization via GBDT: Subspace and RSSI[J]. IEEE Access, 2019, 7: 80519-80529.
23. Ju Y, Sun G, Chen Q, et al. A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting[J]. IEEE Access, 2019, 7: 28309-28318.
24. Shen S, Liu Q, Tao X, et al. Application of the XGBOOST on the Assessment of Transient Stability of Power System[C]//2019 International Conference on Electronical, Mechanical and Materials Engineering (ICE2ME 2019). Atlantis Press, 2019.
25. Chengcheng, Xu, Zixi, et al. CSI-based Autoencoder Classification for Wi-Fi Indoor Localization[C]//Chinese Control and Decision Conference, 2016.
26. Song X, Fan X, Xiang C, et al. A novel convolutional neural network based indoor localization framework with WiFi fingerprinting[J]. IEEE Access, 2019, 7: 110698-110709.