Research Article

Support Vector Regression Based Indoor Location in IEEE 802.11 Environments

Ke Shi, Zhenjie Ma, Rentong Zhang, Wenbiao Hu, and Hongsheng Chen

School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

Correspondence should be addressed to Ke Shi; keshi@mail.hust.edu.cn

The wide spread of the 802.11-based wireless technology brings about a good opportunity for the indoor positioning system. In this paper, we present a new 802.11-based indoor positioning method using support vector regression (SVR), which consists of offline training stage and online location stage. The model that describes the relations between the position and the received signal strength (RSS) of the mobile device is established at the offline training stage by SVR, and at the online location stage the exact position is determined by this model. Due to the complex indoor environment, RSS is vulnerable and changeable. To address this issue, data filtering rules obtained through statistical analysis are applied at offline training stage to improve the quality of training samples and thus improve the quality of prediction model. At the online location stage, k-times continuous measurement is utilized to obtain the high quality RSS input, which guarantees the consistency with the training samples and improves the position accuracy of mobile devices. Performance evaluation shows that the proposed method has a higher positioning accuracy compared with the probability and neutral network method, and the demand for the storage capacity and computing power is also low at the same time.

1. Introduction

Location awareness is one of the keys to the success of pervasive computing. All kinds of pervasive applications such as behavior recognition, intelligent medication, and intelligent building must get the accurate position information of the mobile devices/users to provide accurate and timely services for these devices/users [1–3].

Nowadays, GPS [4] dominates the market of location system. However, GPS cannot function well in the indoor environments because its radio signal suffers from the influence of the buildings, walls, and so forth. In order to provide indoor positioning service, many researchers have proposed a variety of indoor positioning systems based on the principle of different technologies. IEEE 802.11-based positioning system is one of the most successful systems because the 802.11-compliant hardware is inexpensive and already available in many buildings. Existing research results have shown impressive position accuracy.

Many IEEE 802.11-based indoor positioning systems utilize a so-called fingerprinting approach, which uses a two-stage mechanism, offline training phase and online position determining phase. During the offline phase, the signal strength distributions collected from the Wireless Access Points (APs) at predefined reference points in the operation area are stored together with their physical coordinate in a database. During the online position determination phase, mobile devices sample the signal strength of the APs in their communication range and search for similar patterns in the database. The closest match is selected and its coordinate returned to the mobile device as the position estimated.

IEEE 802.11 uses the industrial, scientific, and medical (ISM) band at 2.4 GHz or 5 GHz, and the predominant ones are the 2.4 GHz based substandards called 802.11b/g/n. 2.4 GHz is the resonance frequency of water. This is the reason the human body containing water can disturb the transmissions of these radio signals. Moreover, due to the complex indoor environment, wireless channel congestion, obstructions, and limitation of node communication range, the received signal strength (RSS) is vulnerable and changeable in the practical application. The accuracy of the positioning algorithm seriously suffered from this vulnerable and changeable RSS information. In addition, most of the existing positioning methods require mobile devices to store

the fingerprints, which makes higher storage capacity and computing power of mobile devices in charge in the scene of large-scale applications.

In this paper, we present a novel IEEE 802.11-based indoor positioning method using support vector regression (SVR), which also consists of offline training stage and online location stage. The accurate position prediction model is achieved at the offline training stage by SVR, and at the online location stage the exact position is determined according to the received signal strength (RSS) of the mobile devices. To address the vulnerable and changeable RSS information issues, corresponding data filtering rules obtained through statistical analysis are applied at offline training stage to improve the quality of training sample and thus improve the quality of prediction model. At the online location stage, k-times continuous measurement is utilized to obtain the high quality input, the online received signal strength, which guarantees the consistency with the training samples and improves the position accuracy of mobile devices. Performance evaluation and comprehensive analysis are done through intensive experiments. And the results show that the method proposed in this paper has a higher positioning accuracy when compared with the probability positioning method and the neutral network positioning method, and the demand for the storage capacity and computing power of the mobile devices is also low at the same time.

The remainder of this paper is structured as follows. A summary of related work is presented in the following section. Section 3 describes our positioning method in detail, including SVR-based location modeling, offline training sample filtering, online k-times continuous measuring, and input preprocessing. The performance evaluation and analysis are given in Section 4. Finally, Section 5 draws the conclusions.

2. Related Works

2.1. Existing Position Methods in IEEE 802.11-Based Indoor Positioning Systems. The fingerprinting approach is the most common used location method in 802.11-based indoor positioning systems. Varieties of existing methods have been proposed to find the closest match in the fingerprints, which include k-nearest neighbor (KNN) [5, 6], smallest m-vertex polygon method [7], probabilistic method [8], neural networks [9], and decision tree [10].

In the KNN method proposed by Bahl and Padmanabhan [5], fingerprints are collected by deploying the mobile devices at each predefined reference point in the operation area. Then k-nearest position is found based on the least squares method during the online positioning phase. At last, the location of the target is determined as the average of these k-nearest positions. To improve the accuracy of the KNN positioning method, Bhasker et al. [6] considered the users' feedback during the online location period. The Smallest m-vertex polygon method [7] is similar to the KNN method. They are the same during the offline training phase. During the online positioning period, the smallest m-vertex polygon method takes the location unit in which the signal strength value is

closest to the signal value in each of the m-dimensional signal vectors collected by the real-time detection as an optional vertex, selects the smallest perimeter polygon among all of the m-vertices polygons, and averages these m positions as the predicted results.

The probabilistic method [8] divides the entire positioning scene into a series of discrete areas, $L_1, L_2, \ldots, L_n$. The RSS distribution information of each area is obtained through the training (usually stored in a histogram or distribution of the Gaussian model) during the offline phase. During online positioning phase, S, the observed signal strength vector, is obtained, and the location is determined by the following rule: if

$$P\left(L_i \mid S\right) > P\left(L_j \mid S\right) \quad \text{for } i, j = 1, 2, 3, \ldots, n, \ j \neq i, \quad (1)$$

($P(L_i \mid S)$ is the probability that the signal vector S is observed, given that the mobile node is located in location L_i), then choose L_i as the location of the mobile node. $P(L_i \mid S)$ can be calculated as follows, according to the principle of the Bayesian posterior distribution:

$$P\left(L_i \mid S\right) = \frac{P\left(S \mid L_i\right) P\left(L_i\right)}{P\left(S\right)}, \quad (2)$$

where $P(L_i)$ is the probability of the mobile node in location L_i, and it is usually assumed that the probability of the mobile node at each area follows a uniform distribution in practical application, and $P(S \mid L_i)$ can be calculated according to the RSS distribution information of each area stored during the offline phase, $P(S) = \sum_{i=1}^{n} P(S \mid L_i)P(L_i)$. However, this method can only obtain the discrete position information. Therefore, the continuous position information of the mobile device can only be obtained by interpolation processing.

Mehmood et al. proposed an indoor positioning system based on artificial neural network [9], which also includes offline phase and online phase. During the offline training phase, the trained neural network was obtained through the ⟨coordinates of the location cell, RSS⟩ data. During the online phase, the RSS information collected by the mobile device is taken as the input to the trained neural network and the positioning results are the output of this trained neural network. Compared with the probability method, this method has a high positioning accuracy; the experimental results show that the average error distance decreased from 2.54 meters to 1.43 meters.

Yim [10] builds a decision tree with training data during the offline phase and determines a user's location referring to the tree. This method is more efficient than commonly used probabilistic method, neural networks, and KNN method. Moghtadaiee and Dempstera [11] also used fingerprinting method to implement the indoor location with the FM radio signal. Oussalah et al. [12] proposed a multivariable fuzzy inference system for fingerprinting location, which makes use of k-nearest neighbor classification in signal space and determines target's location as a weighted combination of nearest fingerprints. The weights are determined using Takagi-Sugeno fuzzy controller.

For the methods proposed in the literature [5–8, 10–12], the mobile devices need to store the entire fingerprints, which put a large burden on the mobile devices with limited storage capacity. At the same time, these methods take much time to search the entire fingerprints during the online phase. The method based on neural network [9] has large computational overhead, and the location algorithm is done by the server, not the mobile device, so it leads to transmission delay and may make the server become a bottleneck. It also causes the concern about leaking the user privacy. For most of these methods, the average distance errors are about 2-3 m, and only few of them reduce this value to 1-2 m.

Talvitie et al. [13] considered having an incomplete fingerprint database with realistic coverage gaps and utilized interpolation and extrapolation methods for recovering the missing fingerprint data. The idea of improving the quality of fingerprinting data is similar to our methods. However, the circumstances are different.

2.2. Support Vector Machine Based Positioning Method. The learning methods based on kernel function [14, 15] made tremendous achievements in aspects such as the data classification, the regression estimation, and the function approximation. Support vector machine (SVM) is a new type of machine learning methods among such methods, which has good features such as higher fitting accuracy, fewer parameters, and global optimality. It was generally believed that the data has better separability when the data is extended to high-dimensional feature space, which is suitable for the use of support vector classification, support vector regression, and other kernel function learning methods [16]. Support vector machine includes support vector classification (SVC) and support vector regression (SVR). SVR is used to handle function regression problems. It has been successfully applied to the identification systems, nonlinear forecast systems, and achieved good results.

Battiti [17, 18] made a comparative analysis of various location methods based on statistical learning theory (the results show that the method based on support vector machine has the best results) and recommended the use of support vector classification for location. Nguyen et al. [19] proposed the use of the support vector machine method for location in ad hoc networks. This approach assumes that all nodes can receive the signals of others, there are some anchor nodes whose position was known in the network, and the location of the nodes is divided into n categories. The data collected by the anchor nodes is taken as support vector machine training dataset, and the classification model is then obtained and used to deduct the location during the online phase. Tran and Nguyen [20] studied the wireless sensor network position estimation based on support vector machine and analyzed the upper bound of the positioning error and proposed a modified version of mass-spring optimization to further improve the location estimation. Feng et al. [16] did similar work and proposed a hierarchical support vector machine (H-SVM) program to carry out large-scale sensor network location and carried out an analysis of the average error and variance, as well as the probability distribution of the

positioning error. All these researches [16, 17, 19, 20] studied the location problem in ad hoc and sensor networks based on support vector classification. The scenarios are different with indoor location applications. And the location results correspond to the discrete class, so the location accuracy is closely related to the method and degree of discretization. For example, the method proposed by Tran takes the midpoint of the predicted class and the adjacent class as the location result.

Wu et al. [21] investigated the location estimation for Global System for Mobile (GSM) communication based on support vector regression. Using support vector regression, they studied location estimation problem with the missing value by providing theoretical and empirical analysis on existing and novel kernels. A novel synthetic experiment was designed to compare the performances of different location estimation approaches. The proposed support vector regression approach shows promising performances, especially in terrains with local variations in environmental factors. However, it is for the cellular communication system, not for the 802.11-based wireless local area network.

Space partition methods [22, 23] were presented to improve the location accuracy by dividing the physical space of the target region into small areas according to the RSS features. For each divided area, corresponding SVM model was trained and applied in the location to ease the effects of variable signal. However, even in the small area, the signal may also be changed due to the reasons such as people's movement or network congestion.

Like space partition, Deng et al. [24] also clustered the whole radio map into a series of sub-radio maps by K-means clustering. After that, kernel direct discriminant analysis (KDDA) is used to implement nonlinear discriminative feature extraction of RSS. Then, the relationship between extracted features and physical locations is established by SVR. The focus is to design feature extracting methods that can characterize and capture the nonlinear RSS patterns well.

Our focus is to design appropriate data filtering procedures based on the features of RSS information in IEEE 802.11 environment for training dataset to eliminate the abnormal data and get the location model with high quality. The preprocessing procedure for input data is also included to get the more stable input. The ultimate goal is to achieve higher location accuracy.

3. SVR-Based Location Method

3.1. Location Process. The core of the SVR-based location method is to identify the relationship between the RSS information collected by the mobile device and its geographical location based on support vector regression. The framework of the method is shown in Figure 1. Mobile device obtains its location information through the RSS information collected from 802.11 APs deployed at the predefined location. The entire location process consists of two stages, the offline training stage and the online location stage.
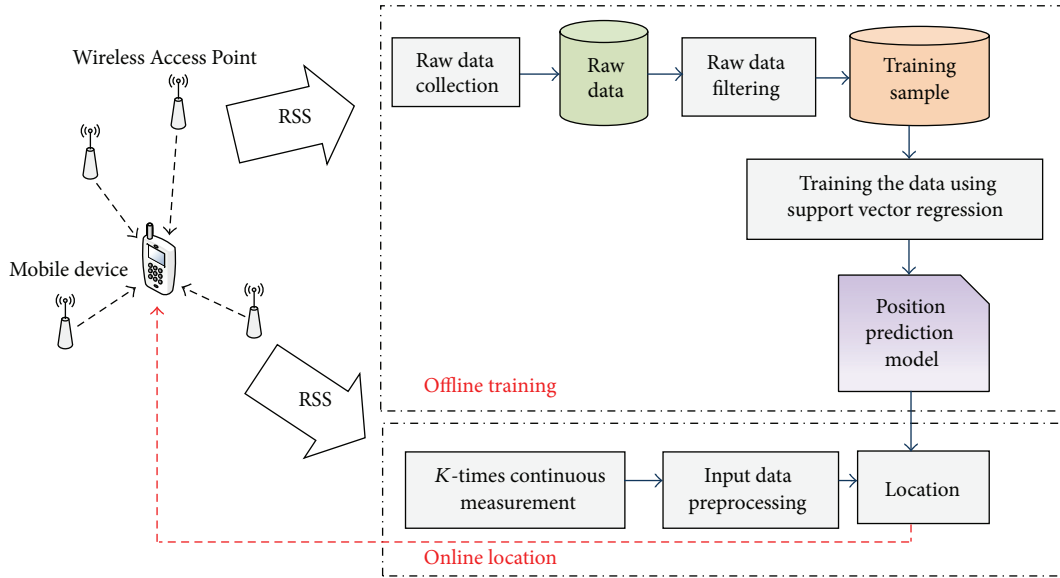
FIGURE 1: The framework of the location algorithm.

3.1.1. Offline Training. During offline training phase, training data is collected and location model is constructed based on SVR. The process is shown as follows.

(i) Raw Data Collection. The mobile device is placed at each predeployed reference point to collect the RSS information from the APs.

(ii) Raw Data Filtering. Due to the complex indoor environment, wireless channel congestion, obstructions, and limitation of node communication range, the received signal strength (RSS) is vulnerable and changeable. To address the above issues, corresponding data filtering rules obtained through statistical analysis are applied to improve the quality of training sample and thus improve the quality of the location model.

(iii) Training the Data Using Support Vector Regression. In this stage, all the collected data is stored on a central server, and data filtering and training are also done on this server. After the appropriate location model is established by training the filtered data using support vector regression algorithm, the mobile devices get this model from this server.

3.1.2. Online Location. During online location phase, mobile devices collect the RSS information from APs and use this information as the input to the location model established in the offline training period to get the real location. The process is shown as follows.

(i) k-Times Continuous Measurement to Collect RSS Information. The RSS information collected by the mobile device is vulnerable, changeable, and inconsistent even at the same position. To get the accurate RSS information, mobile devices execute k continuous measurement and get k RSS values at each location during online phase.

(ii) Input Data Preprocessing. Input data preprocessing analyzes k RSS values obtained through k-times continuous measurement, eliminates the effects of environmental disturbance on the RSS values, and determines the final RSS value that describes the relationships more accurately.

(iii) Location. The RSS information after preprocessing is taken as the input to the position prediction model trained during offline phase and the final location is calculated.

In this stage, mobile devices do not need to exchange the information with the central server. This is because our SVM model is small enough to be stored in the mobile device. And unlike the common fingerprints based methods, there is no need to search all the reference points to match the location.

Offline data filtering and online data preprocessing can effectively reduce the impact of transient variations of RSS values on the location accuracy. However, when major or permanent changes happen, for example, the location of AP changes or the indoor layout changes, the position prediction model needs to be retrained by the latest collected data. This retrained process can be triggered by the location error analysis. If the location error exceeds predefined threshold, the retraining begins.

3.2. Support Vector Regression Based Prediction Model. As shown in Figure 1, the position prediction model describes the relationship between the physical position of the mobile device and the RSS information received from each of the 802.11 APs. Assume that there are n 802.11 APs, and $x_i = \{\mathrm{mac}_1(s), \mathrm{mac}_2(s), \ldots, \mathrm{mac}_n(s)\}$ denotes the RSS information received by the mobile device at the physical location s for each measurement, where $\mathrm{mac}_i(s)$ is the RSS information

received by the mobile device at the physical position s from the ith 802.11 AP. L_i denotes the coordinates of the mobile device in the physical position s. Given a training dataset $D := \{(x_i, L_i)\}_{i=1}^{N}$ $(x_i \in R', L_i \in R)$, the goal of the function regression is to find the mapping $f : R' \rightarrow R$ and make $f(x_i) \approx L_i$. This mapping relationship is nonlinear. For nonlinear problems like this, support vector machine method [21, 25] maps the original data x into a higher-dimensional feature space by utilizing a nonlinear function $\phi(x)$ and then performs linear regression in the feature space

$$L = f(x) = W^T \phi(x) + b, \tag{3}$$

where $\phi(x)$ denotes a nonlinear mapping function from the input space x to a high-dimensional feature space and w and b are the support vector weight and the bias, respectively. The nonlinear regression model is turned into a linear regression model after the input vector x is mapped into the feature space. Parameters w and b are calculated by minimizing the following regularized risk function:

$$R(w) = \frac{1}{2} w^T w + \lambda \sum_{i=1}^{N} \left| L_i - \left(W^T \phi(x) + b \right) \right|_{\varepsilon}. \tag{4}$$

Here, λ and ε are empirical parameters and $|L_i - (W^T \phi(x) + b)|_{\varepsilon}$ represents the ε-insensitive loss function [25]

$$\left| L_i - \left(W^T \phi(x) + b \right) \right|_{\varepsilon}$$
$$= \begin{cases} 0 & \left| L_i - \left(W^T \phi(x) + b \right) \right| < \varepsilon \\ \left| L_i - \left(W^T \phi(x) + b \right) \right| - \varepsilon, & \left| L_i - \left(W^T \phi(x) + b \right) \right| \geq \varepsilon. \end{cases} \tag{5}$$

The value of the loss function is 0 when the predictive value of the error is less than ε; otherwise, linear punishment is applied.

By introducing the positive slack variables ξ_i and ξ_i', the minimization of (4) is equivalent to minimizing the following constrained risk function:

$$R(w) = \frac{1}{2} w^T w + C \left(\sum_{i=1}^{N} \left(\xi_i + \xi_i' \right) \right) \tag{6}$$

subject to the constraints

$$L_i - W^T \phi(x_i) \leq \varepsilon + \xi_i,$$

$$W^T \phi(x_i) - L_i \leq \varepsilon + \xi_i', \tag{7}$$

$$\xi_i \geq 0, \ \xi_i' \geq 0, \ i = 1, \ldots, N,$$

where C is a regularization constant, controlling a compromise between maximizing the margin and minimizing the number of training set errors, and ξ_i and ξ_i' represent upper and lower constraints on the outputs of model.

This constrained optimization problem can be defined as the following Lagrangian function:

$$L\left(w, \xi_i, \xi_i', \alpha, \alpha', t, t'\right)$$
$$= \frac{1}{2} w^T w + C \left(\sum_{i=1}^{N} \left(\xi_i + \xi_i' \right) \right) - \sum_{i=1}^{N} \left(t_i \xi_i + t_i' \xi_i' \right)$$
$$- \sum_{i=1}^{N} \alpha_i \left(W^T \phi(x_i) - L_i + \varepsilon + \xi_i \right)$$
$$- \sum_{i=1}^{N} \alpha_i' \left(L_i - W^T \phi(x_i) + \varepsilon + \xi_i' \right), \tag{8}$$

where α, α', t, and t' are the Lagrange multipliers. The optimization of $L(w, \xi_i, \xi_i', \alpha, \alpha', t, t')$ must satisfy the following conditions:

$$\frac{\partial L\left(w, \xi_i, \xi_i', \alpha, \alpha', t, t'\right)}{\partial w} = w - \sum_{i=1}^{N} \left(\alpha_i - \alpha_i' \right) \phi(x_i)$$
$$= 0,$$

$$\frac{\partial L\left(w, \xi_i, \xi_i', \alpha, \alpha', t, t'\right)}{\partial \xi_i} = t_i - \left(C - \alpha_i \right) = 0, \tag{9}$$

$$\frac{\partial L\left(w, \xi_i, \xi_i', \alpha, \alpha', t, t'\right)}{\partial \xi_i'} = t_i' - \left(C - \alpha_i' \right) = 0.$$

The above problem can be transformed to dual optimization problem by substituting (9) into (8). A convex function can be obtained:

$$Q\left(\alpha, \alpha'\right) = \sum_{i=1}^{N} L_i \left(\alpha_i - \alpha_i' \right) - \varepsilon \sum_{i=1}^{N} \left(\alpha_i + \alpha_i' \right)$$
$$- \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left(\alpha_i - \alpha_i' \right) \left(\alpha_j - \alpha_j' \right) K\left(x_i, x_j\right). \tag{10}$$

The solution can be obtained by maximizing (10) subject to a new set of the constraints:

$$\sum_{i=1}^{N} \left(\alpha_i - \alpha_i' \right) = 0,$$
$$0 \leq \alpha_i \leq C, \ 0 \leq \alpha_i' \leq C, \ i = 1, \ldots, N. \tag{11}$$

With the Lagrange multipliers α_i and α_i', the estimated output can be represented by

$$L = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i' \right) K\left(x, x_i\right) + b, \tag{12}$$

where $K(x, x_i)$ is called the kernel function. According to Karush-Kuhn-Tucker's (KKT) conditions of solving quadratic programming problem, only some of the coefficients, $\alpha_i - \alpha_i'$, are not zeros, and the corresponding training
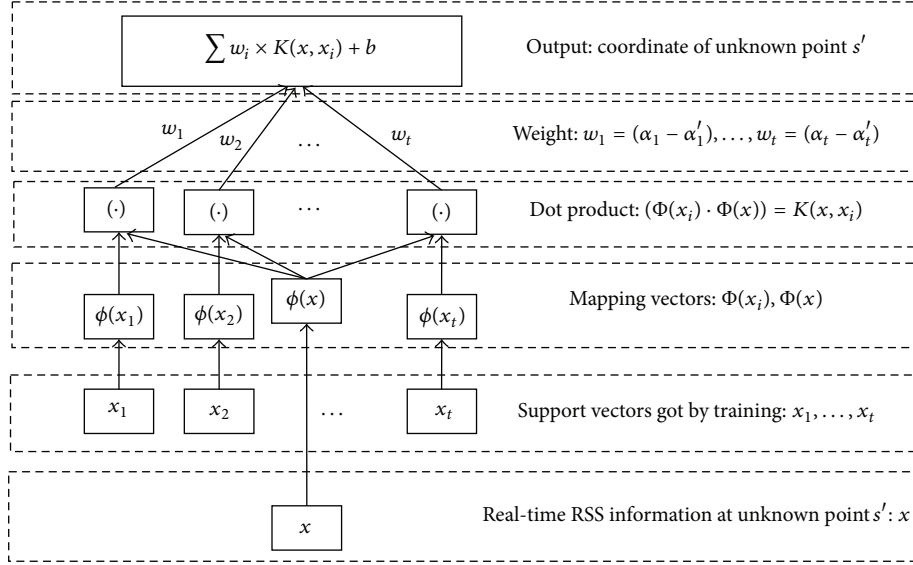
FIGURE 2: The location prediction model obtained by SVR.

data vectors are called support vectors. The final prediction model trained by SVR is shown in Figure 2.

In support vector classification and regression, a well selected kernel function determines how well the input data are mapped. Commonly used kernels include

(i) linear kernel

$$K\left(x_i, x_j\right) = x_i^{T} \cdot x_j, \tag{13}$$

(ii) polynomial kernel

$$K\left(x_i, x_j\right) = \left(\gamma\left(x_i^{T} \cdot x_j\right) + c\right)^{d}, \tag{14}$$

(where d is the degree of the polynomial and γ and c are constants),

(iii) Gaussian RBF kernel with the following form:

$$K\left(x_i, x_j\right) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{\sigma^2}\right), \tag{15}$$

(where σ is a user-specified parameter and $\|\cdot\|$ denotes the distance between two input vectors),

(iv) Sigmoid kernel

$$K\left(x_i, x_j\right) = \tanh\left(\gamma\left(x_i^{T} \cdot x_j\right) + \beta\right), \tag{16}$$

(where γ and β are constants).

3.3. Data Filtering during the Offline Phase. The quality of the training sample determines the quality of the final position prediction model when using the support vector regression algorithm: the better the quality of the training sample is, the better the final model is. Meanwhile, the more precise the input of the model is, the more accurate the location

results obtained are. In the wireless indoor environment, the wireless signal transmission is influenced by many factors, and the RSS information received by the mobile device is variable due to complex transmission environments. Even at a fixed point, the RSS information collected by the mobile device will be changing, which may compromise the quality of the training sample and the model input and affect the final position accuracy. Therefore, in order to obtain the high quality training samples, the raw data needs to be filtered at the offline phase. And the input RSS information must also be preprocessed to obtain a more accurate input at the online phase. Therefore, we will discuss the raw data filtering rules in this section and the input RSS information preprocessing in the next section.

There are a lot of existing methods to ease the effects of abnormal data. For example, averaging is a simple and common used method, which can give a measure that is more robust in the presence of outlier values. Raw data filtering can be done through the simple averaging of the RSSI value groups. For wireless environment, reliability is always a concern, which makes the packets containing RSS information vulnerable to loss and leads to lower RSS value. Based on this intuition, the rule of removing the least values eliminates certain number of the lowest values from a RSSI value group.

However, 802.11 radio transmissions have their own characteristics. To get better result, filtering rules should be designed according to these characteristics. Through an analysis of the collected data, the probability distribution of the RSS information collected by the mobile device at a fixed point from the APs is shown in Figure 3; we found that it is consistent with the log-normal distribution law described in [26, 27].

Assume

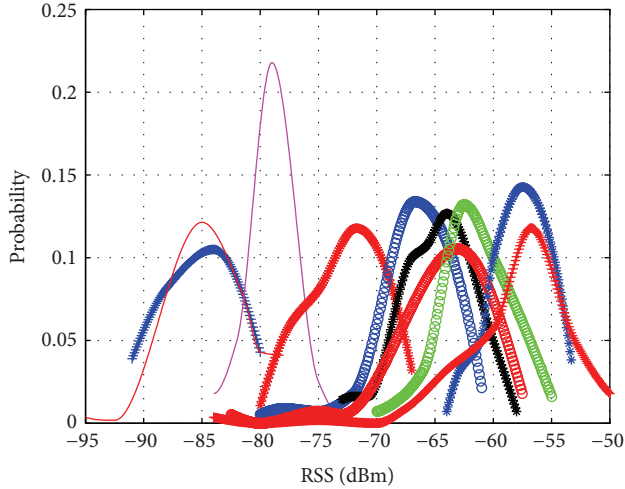$$x = -\text{mac}_i\left(s\right) > 0. \tag{17}$$

FIGURE 3: The distribution of the RSS information.

Then, the probability density function of x is as follows:

$$f(x) = \frac{1}{\sqrt{2\pi} \times \sigma \times x} \times \exp^{-(\ln x - \mu)^2 / 2\sigma^2}, \qquad (18)$$

where $\mathrm{mac}_i(s)$ denotes the RSS information received by the mobile device in the physical location s from the ith 802.11 AP, μ is the logarithmic mean of x, and σ is the logarithmic standard deviation of x.

Then, the expected value of x is

$$E(x) = e^{\mu + \sigma^2 / 2}. \qquad (19)$$

Based on the probability distribution, we can set a positive threshold θ, if and only if $0 < E(x_i) - \theta < -\mathrm{mac}_i(s) < E(x_i) + \theta \, \mathrm{mac}_i(s)$ is considered as a valid value. And if the value is beyond the range, too high or too low, it is considered as an abnormal signal value. For each measurement value $x = -\mathrm{mac}_i(s)$, the probability of a valid value is

$$P_V\left(E(x_i) - \theta < x < E(x_i) + \theta\right)$$
$$= \int_{E(x_i)-\theta}^{E(x_i)+\theta} \frac{1}{x \times \sigma \times \sqrt{2\pi}} \times \exp\left(-\frac{(\ln x - \mu_i)^2}{2\sigma^2}\right) dx. \qquad (20)$$

In addition to the abnormal RSS values, due to network congestion and lossy wireless links, the packet containing the RSS information from AP may be lost, and the mobile device may get a record with $\mathrm{mac}_i(s) = 0$. Even in a good transmission environment, high network load would cause packet dropping. Therefore, we need to distinguish the situation where the packet loss makes $\mathrm{mac}_i(s) = 0$ from the situation where the positional relationship between the mobile device and the corresponding 802.11 APs makes the signal intensely weak. Therefore, utilizing multiple measurements to distinguish between these two cases is proposed.

Assuming that measurements of the device are independent of each other, for each reference point, the probability

that, in each measurement, the mobile device cannot receive the RSS information from the ith AP at this position is

$$P_{Mi} = \frac{m - j}{m}, \qquad (21)$$

where m is the total number of measurements that the mobile device made at the reference point and j is the total number of measurements where the mobile device received the RSS information from the ith AP at the reference point.

If P_{Mi} is very high, it suggests that the mobile device is far from the ith AP and the received signal strength itself is low, or the communication between the mobile device and the ith AP is interfered in (such as barriers); the RSS information is unstable. In such cases, even multiple measurements cannot obtain a valid signal strength value. If P_{Mi} is very low, it suggests that the device is within the communication range of the ith AP; it is able to receive a normal signal strength value. In this situation, $\mathrm{mac}_i(s) = 0$ is caused by the drop of the packet containing the signal strength information due to the reason such as network congestion. In order to improve the accuracy of the location prediction model, these data records should be removed from the training sample. Therefore, we define a valid measurement threshold ζ. If $P_{Mi} \leq \zeta$, we believe that the mobile device at the reference point can communicate effectively with the ith AP, and most measurements can obtain the valid values; if $P_{Mi} > \zeta$, we believe that the communication between the mobile device and the ith AP is subject to greater interference, and the mobile device cannot get the signal strength information; most measurements are invalid. Based on the above analysis, we define the corresponding data filtering rules as follows during the offline period:

(1) For each record collected by the mobile device at the reference point from the APs, if $P_{Mi} \leq \zeta$ for $i = 1, 2, 3, \ldots, n$ (where ζ is a valid measurement threshold value) and $\mathrm{mac}_i(s) = 0$, then the record is considered to be invalid and will be removed; if $P_{Mi} > \zeta$ for $i = 1, 2, 3, \ldots, n$ and $\mathrm{mac}_i(s) \neq 0$, it is also considered to be invalid and will be removed as well.

(2) For each record collected by the mobile device at the reference point from the APs, if $P_{Mi} \leq \zeta$ for $i = 1, 2, 3, \ldots, n$ and $-\mathrm{mac}_i(s) < E(x_i) - \theta$ or $-\mathrm{mac}_i(s) > E(x_i) + \theta$, the record is considered to be invalid and will be removed.

The first case of the first rule describes the situation that the packet containing the normal signal strength is lost due to the network congestion; the record should be removed; the second case describes the situation where a small amount of unreliable records generated when the device and AP are in the margin of their communication range should be filtered out. The second rule describes the situation where the abnormal signal should be filtered out when the communication between the mobile device and the APs was disturbed.

3.4. Input Data Preprocessing during Online Phase. The unstable and abnormal data are removed from the training sample

after the offline period, and an accurate position prediction model is obtained. To get the position information with high accuracy, the input to the model during online phase should also be of high accuracy; unstable and abnormal RSS data should be avoided too. k-times continuous measurements to collect RSS information are proposed. During the online period, the mobile device performs k continuous measurements at first to ensure that the valid RSS information can be acquired; then the input data preprocessing method removes the abnormal and disturbed data from the obtained data to get high quality input that truly reflects the relationship between the mobile device and the 802.11 APs.

If the simple rules such as averaging and removing the least values are used in the data filtering, the same rules can also be used in the data preprocessing. For the sophisticated data filtering rules we discussed in the previous subsection, more complex preprocessing rules are needed. Under this circumstance, to obtain the valid RSS information, we must set an appropriate k value.

Assume

$$x = -\text{mac}_i\left(s'\right) > 0, \tag{22}$$

where $\text{mac}_i(s')$ denotes the RSS information received by the mobile device at the physical location of s' from the ith 802.11 AP.

Based on the above analysis, x follows a log-normal distribution. Let $P_V(E(x_i) - \theta < x < E(x_i) + \theta)$ denote the probability that x is in the range $(E(x_i) - \theta, E(x_i) + \theta)$ (where $E(x)$ denotes the expected value of x and $E(x_i) - \theta > 0$). When considering the situation where the signal strength cannot be properly received by the mobile device, let $P(E(x_i) - \theta < x < E(x_i) + \theta)$ denote the probability that the RSS value received by the mobile device in each measurement is in the range $(E(x_i) - \theta, E(x_i) + \theta)$ (where $E(x_i) - \theta > 0$); then

$$P\left(E\left(x_i\right) - \theta < x < E\left(x_i\right) + \theta\right)$$
$$= \left(1 - P_{Mi}\right) \times P_V\left(E\left(x_i\right) - \theta < x < E\left(x_i\right) + \theta\right), \tag{23}$$

where P_{Mi} denotes the probability that, in each measurement, the mobile device cannot receive the RSS information from the ith AP at this position.

In order to ensure that the mobile device is able to obtain the valid RSS information after the k-times continuous measurements during the online period, the following condition must be satisfied:

$$1 - \left(1 - P\left(E\left(x_i\right) - \theta < x < E\left(x_i\right) + \theta\right)\right)^k = \tau, \tag{24}$$

where τ is a constant approaching 1.

Then,

$$k = \log_{\left(1 - P\left(E(x_i) - \theta < x < E(x_i) + \theta\right)\right)}\left(1 - \tau\right). \tag{25}$$

In the real system, the value of k must be greater than or equal to the derived value from formula (25). The larger the value of k is, the more the amount of the valid RSS information obtained by the mobile device is. At the same time, the larger k value also means that the mobile device

takes more time to measure, which leads to longer delay and deteriorates the user experience. The scope of application may also be restricted; for example, the moving speed of the user is limited. Formula (25) gives a lower limit of the k value. The real value of k can be determined based on the indoor wireless environment and the needs of users. In general, the better the wireless transmission environment is, the lighter the wireless network load is, the less the packet loss is, the lower the probability of the signal's abnormalities is, and the smaller the k value can be. On the other hand, if the requirements of users on location accuracy are low, the k value can also be smaller. From formula (25), it can be seen that the lower limit of the k value is directly related to the parameters θ and ζ. θ reflects the range of the users' tolerance to the signal strength changes, and ζ reflects the reliability of the assurance. The optimal k value should be determined by users' need and real environment.

After the value k was determined, the mobile device gets the valid RSS information through k-times continuous measurements during the online period. Then, this information is preprocessed as follows.

Let

$$x_i = -\text{mac}_i\left(s'\right) > 0. \tag{26}$$

For each AP, P_{Mi} is calculated first. If $P_{Mi} \geq \zeta$, let $\text{mac}_i(s') = 0$. Otherwise, the logarithmic mean μ_i and standard deviation σ_i of x_i are calculated and the expected value $E(x_i)$ of x_i is determined according to (19). If $|-\text{mac}_i(s') - E(x_i)| < \theta$, $\text{mac}_i(s')$ is considered as a valid value, denoted by $\text{mac}_{ij}(s')'$; otherwise, it is considered as an invalid value. The final $\text{mac}_i(s')$ value is the mean of all valid values:

$$\text{mac}_i\left(s'\right) = \begin{cases} 0 & P_{Mi} \geq \zeta \\ \dfrac{\sum_{j=1}^{t} \text{mac}_{ij}\left(s'\right)'}{t} & P_{Mi} < \zeta, \end{cases} \tag{27}$$

where ζ is a valid measurement threshold value, $\text{mac}_{ij}(s')'$ is a valid measuring value, and t is the total number of the valid values corresponding to the k-times continuous measurements from the ith AP.

After preprocessing, we can obtain the valid and true RSS information that is consistent with the offline training sample. It serves as the input to the position prediction model, which greatly improves the location accuracy.

4. Experimental Results

4.1. Simulation Environment. The performance of this positioning method is evaluated through simulation. The data comes from the real deployment and experiments of 802.11-based wireless positioning system as shown in Figure 4. The grid of reference spots in the operation area includes 130 spots with spacing of 1.5 meters. Wherein the blue spots represent the position of the reference point, the yellow spots represent the location of the AP, and the red spots represent the test position during online period. The entire system covers an
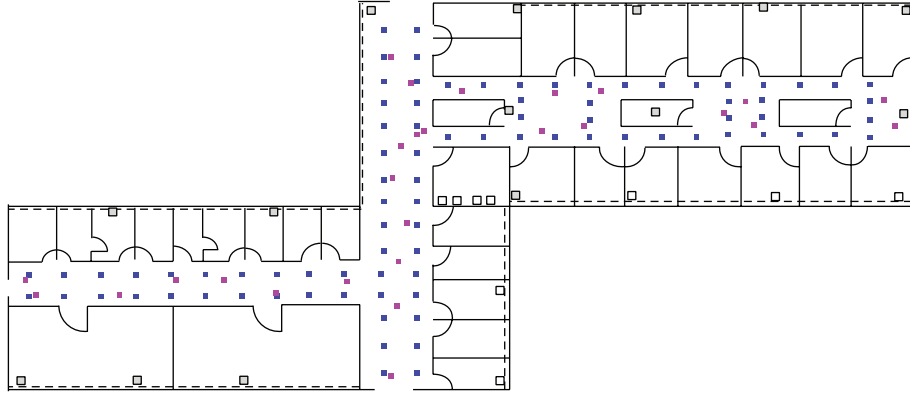
FIGURE 4: The deployment of the positioning system.

area of 221 square meters, and 25 802.11 APs are deployed in the entire region. In our experiments, not all the APs are used. The default number of used APs is 17; that is to say, $n = 17$. We also change the number of used APs to study its effect on the accuracy. And 110 signal strength samples are collected at each reference spot, which means $m = 110$. For each measurement, it takes 250 milliseconds.

The implementation of SVR is based on the LIBSVM; the parameters of the SVR are fixed at an empirically favorable value: $c = 1,000$, $\varepsilon = 0.01$. To measure the positioning error, let $\{x_p, y_p\}$ denote the predicting coordinates of the position prediction model, and $\{x_r, y_r\}$ denote the actual coordinates; the location error distance is calculated as

$$\text{Error} = \sqrt{\left(x_p - x_r\right)^2 + \left(y_p - y_r\right)^2}. \tag{28}$$

In the experiment, the training samples were filtered according to the corresponding filtering rules and the k-times continuous measurement method is used to collect RSS information during the online period. The experimental parameters are described in Table 1.

4.2. The Impacts of the Data Quality on the Location Method.
Our method improves the quality of the training sample and input data by the filtering operation during the offline period and input data preprocessing during the online period. In order to analyze the influence of the quality of the corresponding data on the accuracy of the positioning method based on SVR, four experiments have been done to compare with each other. In experiment 1, the training sample is not filtered during the offline period, and the mobile device only collects 1 sample during the online period at the unknown point. In the second experiment, the training sample is not filtered during the offline period either, but the mobile device repeated 8-time measurements during the online period at the unknown point, and then the records are preprocessed. In experiment 3, the training sample is filtered during the offline period, but the mobile device only collects 1 sample during the online period at the unknown point. In experiment 4, the training sample is filtered during the offline period, and the mobile device repeated 8-time measurements
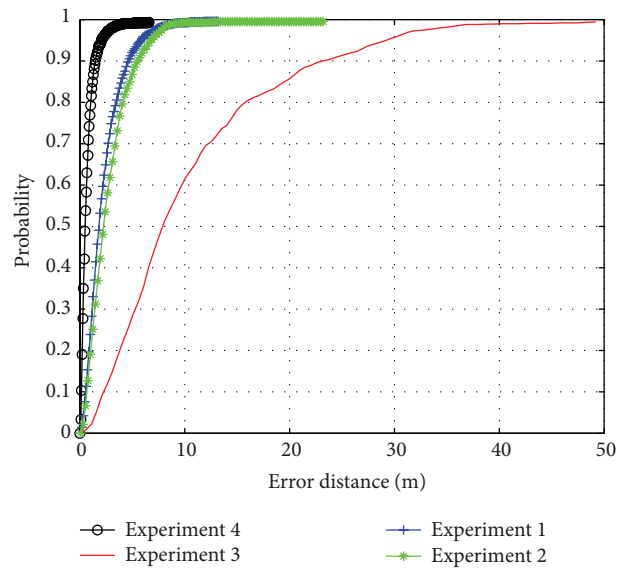

FIGURE 5: Relation between the data quality and error distance.

during the online period at the unknown point, and then the records are preprocessed. Figure 5 shows the cumulative probability distribution (CDF) of the error distance for these four scenarios. The average error distance is shown in Table 2.

It can be seen that the positioning accuracy of experiment 4 is the best, and the average error distance is 0.68 m, less than 1 m. The results of experiment 1 rank second, and the average error distance is 2.25 m. Then experiment 2 follows; the average error distance is 2.70 m. Experiment 3 generates the worst results; the average error distance reaches 10.37 m. Therefore, the filtering operation during the offline period and the k-times continuous measurement to collect RSS information during online period can effectively improve the position accuracy. The results from experiment 1 indicate that original SVR without training data filtering and input preprocessing has good features such as higher fitting accuracy, fewer parameters, and global optimality and can establish good mapping nonlinear relations between RSSI and location. To further improve the performance, better

TABLE 1: Experimental parameters description.

Parameters name	Default value	Description
Training sample size	5000	The default size of the training sample is 5000, while the value is varied in the simulation to reveal the relation between the training sample size and the performance.
The number of APs	17	The default number of APs is 17, while the value is varied in the simulation to reveal the relation between the number of APs and performance.
Kernel function	RBF kernel	The default kernel function is RBF kernel, while different kernel functions are used in the simulation to reveal its impact on the performance.
The value of k	8	The default value of k is 8, while the value is varied in the simulation to reveal its impact on the performance.
ζ	0.9	The default value of ζ is 0.9, while the value is varied in the simulation to reveal its impact on the performance.
θ	4	The default value of θ is 4, while the value is varied in the simulation to reveal its impact on the performance.
Test sample size	1000	

TABLE 2: Error distances in different experiments.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
50% of measurements	1.78 m	2.23 m	7.78 m	0.48 m
80% of measurements	3.41 m	4.02 m	15.60 m	1.00 m
90% of measurements	4.51 m	5.40 m	22.40 m	1.40 m
Average value	2.25 m	2.7 m	10.37 m	0.68 m

prediction model and better input are both needed. Better prediction model with worse input or worse prediction model with better input cannot compete with the original SVR because the model and the input do not fit and twist the mapping relations. This is why the results of experiment 1 are better than the results from experiments 2 and 3. The high quality model trained by high quality does need high quality input data.

Better model combined with worse input leads to the worst result because our model does not search all the reference points and the results depend on the input. Also worse model combined with better input cannot lead to better results. Since the size of training data is huge, the percentage of the abnormal data is small and the effect on the model is relatively small. At the online stage, sampling the abnormal data would make the input completely wrong. This is the reason why the results of experiment 3 are worse than experiment 2.

4.3. Performance Comparison. In order to demonstrate the effectiveness of the location method (denoted as Preprocessing SVR) proposed in this paper, we have it compared with the probabilistic method [26] (denoted as Probabilistic Model), the probabilistic method in which the fingerprinting data has been filtered and the input has been preprocessed (denoted as Preprocessing Probabilistic Model), the SVR method with no filtering and no preprocessing (denoted as No Preprocessing SVR), the ANN method [9] (denoted as No Preprocessing ANN), and the ANN method in which there is the same operation as the Preprocessing SVR method during the offline period and the online period (denoted

as Preprocessing ANN). For each method, the CDF of the final location error distance is shown in Figure 6. The average values are shown in Table 3, from which we can see that the positioning accuracy of the Preprocessing SVR method (our proposed method) is the best among all these methods. The performance of the Preprocessing ANN method is close to the Preprocessing SVR method. The average error distance of the Preprocessing ANN method is 0.886 m. The average error distance of the Preprocessing SVR method is 0.68 m. Without filtering and preprocessing, the average error distance of the No Preprocessing SVR method reaches 2.25 m, and the average error distance of the No Preprocessing ANN method also reaches 2.26 m.

Therefore, the effective data filtering and preprocessing operation can greatly improve the accuracy of the positioning methods based on machine learning; the performance of the SVR methods is superior to the ANN methods. Compared with SVM model, the size of ANN method is larger and its computing complexity is also higher; therefore, ANN methods cannot be implemented on the device side. The mobile devices have to send the sample to the server where the location results are generated and sent back to the devices. ANN methods that are implemented on the server side cause extra delay and extra network overload.

The average error distance of the probability model method is 2.857 m. With preprocessing and filtering, it decreases to 2.19 m. It shows good data quality can improve the performance of probability model. But it still cannot compete with the SVR method. Unlike SVR and ANN methods, the Probabilistic Model does not have sophisticated data training process to get a prediction model. The locations

TABLE 3: Comparison of the error distances of the different location methods.

	Preprocessing ANN	ANN	Preprocessing SVR	No Preprocessing SVR	Probabilistic Model	Preprocessing Probabilistic Model
50% of measurements	0.74 m	1.87 m	0.48 m	1.78 m	2.47 m	2.01 m
80% of measurements	1.25 m	3.17 m	1.00 m	3.41 m	4.10 m	3.61 m
90% of measurements	1.66 m	4.24 m	1.40 m	4.51 m	5.21 m	4.61 m
Average value	0.886 m	2.24 m	0.68 m	2.25 m	2.857 m	2.19 m

- No Preprocessing SVR
- Preprocessing SVR
- No Preprocessing ANN
- Preprocessing ANN
- Probabilistic Model
- Preprocessing Probabilistic Model
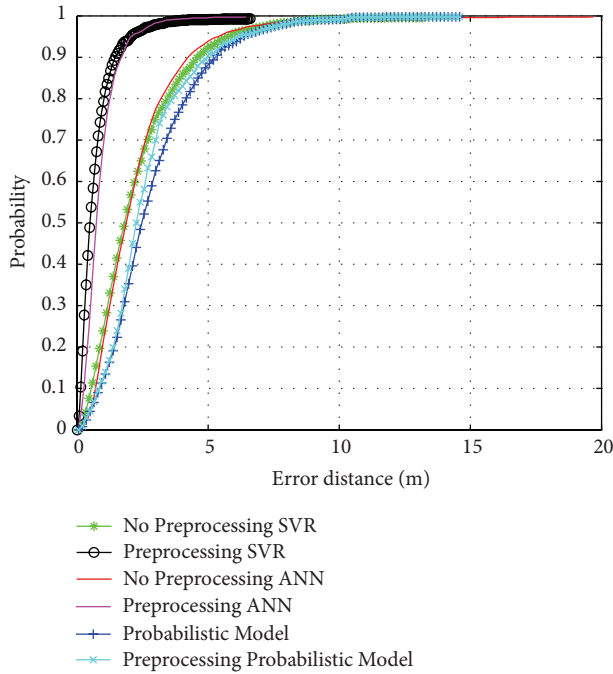
FIGURE 6: CDF of the error distance of different methods.

FIGURE 7: Training sample size versus average error distance.

are determined through a searching and matching process. SVR and ANN can capture more accurate mapping relations between the RSSI information and locations. To the best of our knowledge, SVR and ANN usually perform better than the Probabilistic Model, which have been proved by the existing literatures.

4.4. The Impact of Training Sample Size. The size of the training sample is a critical parameter to the positioning algorithm. It determines the lower bound of the time needed to collect data for the training. To analyze the impact of training sample size on the performance, we vary the training sample size between 1000 and 6000. The corresponding average error distance is shown in Figure 7. The average error distance rapidly reduced as the training sample size increases when the training sample size is less than 2000. When the training sample size is between 2000 and 4000, the average error distance decreases significantly slower. And when the training sample size is greater than 4000, the average error distance is almost constant as the training sample size increases.
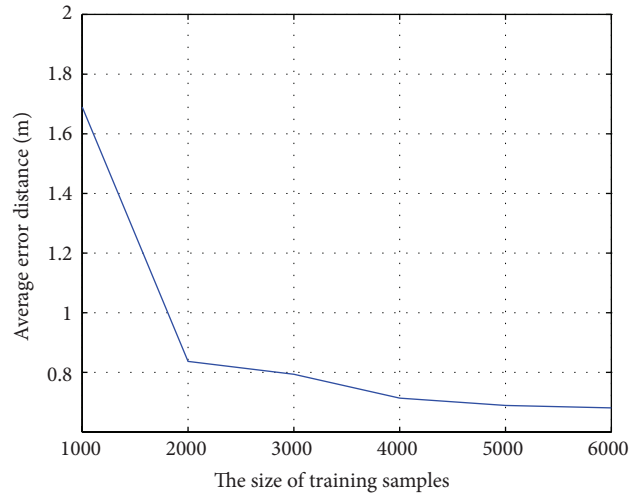
4.5. The Impact of Kernel Function. According to the support vector regression theory, the kernel functions that meet the demands of the Mercer Theorem transform the dot product operation in high-dimensional feature space into the function calculation in low-dimensional feature space [16] to solve the "curse of dimensionality" problem. The location prediction model obtained by different kernel functions will have different performance. In the experiments, we tested the linear kernel, the polynomial kernel, the RBF kernel, and the Sigmoid kernel. The results are shown in Figure 8; we can see that the location prediction model using the RBF kernel has the highest accuracy. Using the linear kernel, the average error distance is 2.44 m. Using the Sigmoid nuclear function (where $\gamma = 1/17$, $\beta = 0$), the average error distance was 2.46 m. The average error distance was 0.68 m with the RBF kernel used. Using polynomial kernel ($\gamma = 1/17$, $C = 0$, and $d = 3$), the average error distance is 1.15 m. In our experiments, we try the different parameters for all the kernel functions. For each kernel function, the results presented in our paper are the best results got in these experiments.

4.6. The Impact of the Number of APs. In general, the large number of APs in a certain region leads to higher location accuracy. In the previous sections, the number of APs is 17; now we change the number of APs from 1 to 20 to study the impact of the number of APs on the location accuracy. The APs used for location are randomly selected from all
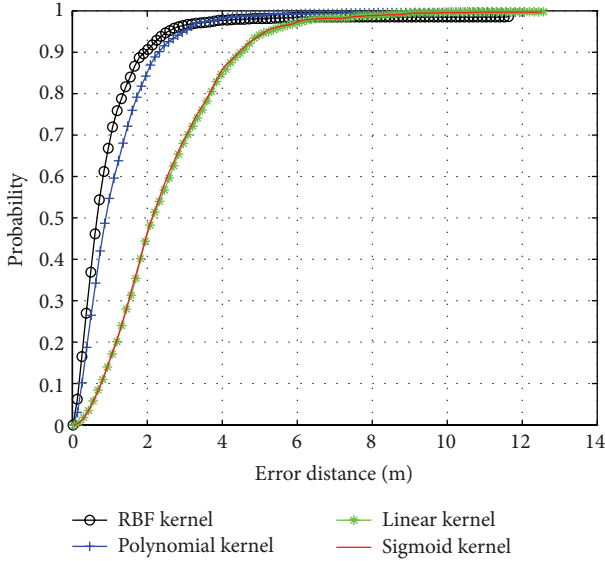
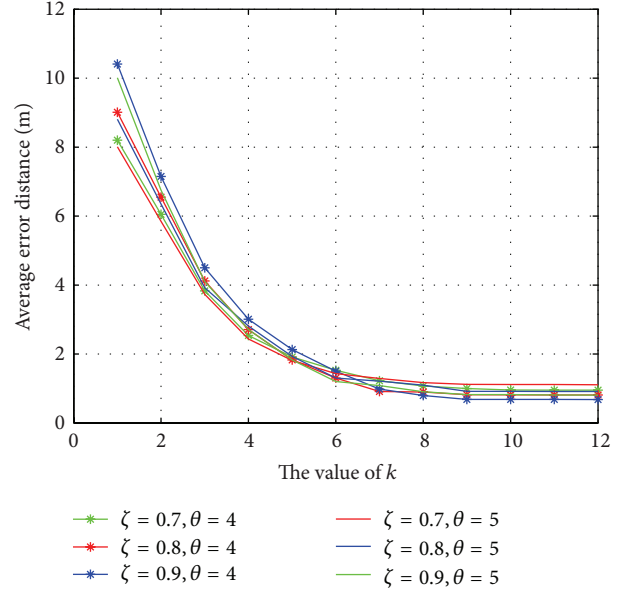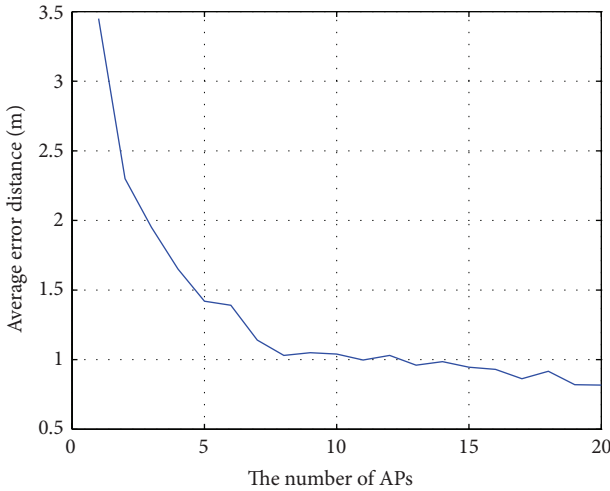FIGURE 8: CDF of the error distance of different kernel functions.

FIGURE 9: The average error distance versus the number of APs.

FIGURE 10: The value of k versus average error distance.

the APs. We repeat the same experiment as mentioned above. The results are shown in Figure 9. These results are consistent with our original prediction that the higher the number of APs is, the higher the accuracy of the system is. When the number of APs is 20, the minimal average error distance is achieved. However, the importance of APs to the location is not the same; the location accuracy does not increase smoothly with the increase of the number of APs. Some APs may have more important influence on the positioning accuracy of the location methods than other APs. Therefore, how to deploy the appropriate number of APs in the right places needs further research. Moreover, when the number of APs reaches a certain number, a high positioning accuracy can be obtained by the location method proposed in this paper, but there is no significant improvement in positioning accuracy after the number of APs exceeds 8.

4.7. Determination of the k Value. As described before, the value of the parameter k has an important impact on the positioning accuracy of the system, and it is closely related to the parameters θ and ζ and the deploying environment. To obtain a good positioning accuracy and meet the users' needs at the same time, k should be set to a specific value according to the wireless transmission environment and the time delay requirements of users, as well as the positioning accuracy. In order to get the relation between the value of k and the location prediction model, we change the value of k from 1 to 12 one by one with corresponding θ and ζ. The experimental results are shown in Figure 10. Overall, the positioning accuracy of the model increases with the increase of the value of k. The values of the parameters θ and ζ identify the training sample filtration standard; the more stringent the standard, the larger the value of k required for the accuracy of positioning model to reach convergence and, at the same time, the better the accuracy. On the contrary, the lower the standard, the less the value of k required for the accuracy of positioning model to reach convergence and the lower the location accuracy.

4.8. Resource Consumption of the Mobile Device. Due to a limited storage space and computing power of the mobile devices, how to reduce the demand for data storage capacity and computing power of the mobile devices for the positioning methods is also very important. In order to reduce the demand for data storage capacity of the mobile devices, the literature [28] proposed an algorithm for 80.11-based positioning system. For this algorithm, the experimental results show that the mobile terminal equipment only needs to store 12 Kbytes of data and greatly reduced the demand for storage capacity of the mobile device. However, the mobile devices need to communicate with the data storage center and update the "fingerprint" data synchronically, which greatly

aggravates the communication loads for the mobile device. For the method proposed in this paper, if the size of training sample is 5000 and the kernel function is the Gaussian kernel function and the number of the APs is 17, the size of the obtained position prediction model is only 81 Kbytes. It is far smaller than the entire "fingerprint" data (6110 Kbytes), which greatly reduced the storage load of the mobile device. The ANN method [9] that generally provides location-based services in server-client mode due to computational complexity causes the problem of users' privacy protection. For our positioning method, the main work of the calculation is completed during the offline period, and it can calculate the location information of the mobile device with constant time complexity during online period; thus the requirements for the computing power of the mobile device are not high.

4.9. The Impacts of Filtering/Preprocessing Rules. Good data quality leads to accurate location results. Filtering/preprocessing rules decide the quality of trained data and input data. In our paper, the particular preprocessing/filtering rules are devised according to the characteristics of 802.11 radio environment. To prove the efficiency of these rules, we compared them with two other filtering/preprocessing methods mentioned in our paper, averaging and removing the least values. The results are shown in Figure 11. With averaging, only very slight improvement is achieved compared with the No Preprocessing SVR. The two curves are almost identical. The reason is that the unreliable wireless transmission causes lower RSS values and has huge impacts on the location results. Removing the least values performs better than averaging because it can remove some unreal lower RSS values. Our approaches perform the best, which indicates that custom-made preprocessing/filtering approaches are necessary to get high accuracy.

We also compare our approach with KDDA [24]. According to the existing literature [24], the mean error is 1.46 m in the experiments, which is higher than Preprocessing SVR and lower than the original SVR. Of course, the comparison may be unfair because the environments and setups are different. Therefore, we implement our own version of KDDA without location clustering and used it as the filtering/preprocessing rules in the location process. As shown in Figure 11, KDDA performs better than averaging and removing the least values, which indicates that mapping RSS signal into a high-dimensional kernel space can help in extracting the signal feature. However, our Preprocessing SVR still performs better because filtering/preprocessing rules improve the data quality significantly through removing the abnormal data samples.

5. Conclusions and Future Work

Location awareness as a part of a context-aware computing paradigm is one of the keys to the success of ubiquitous and pervasive computing. In this paper, we present a new 802.11-based indoor positioning method using support vector regression. Due to the complex indoor environment, wireless channel congestion, obstructions, and limitation of node
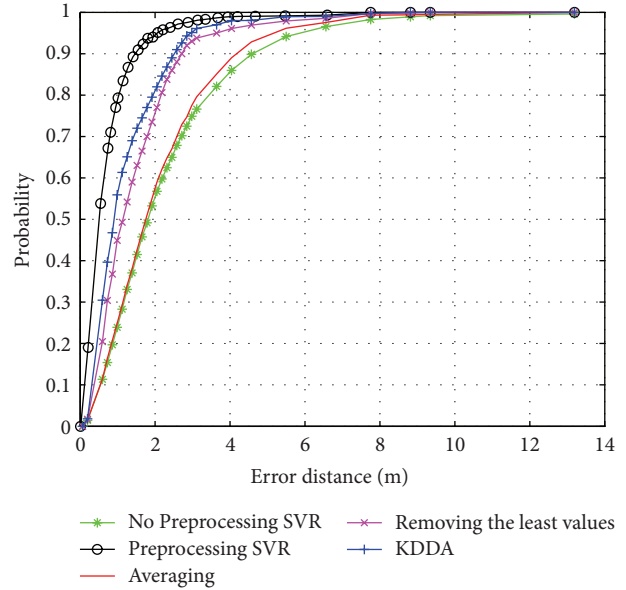

FIGURE 11: CDF of the error distance versus filtering/preprocessing rules.

communication range, the received signal strength (RSS) is vulnerable and changeable. To address the above issues, corresponding data filtering rules and k-times continuous measurement method were proposed. And the results show that the method proposed in this paper has a higher positioning accuracy compared with the probability positioning method and neutral network positioning method. The demand for the storage capacity and computing power of the mobile devices is also low at the same time.

The work of the future research will focus on the analysis of the relationship between the position of the APs and the positioning accuracy of the location method. Given a certain number of APs, how to make full use of them to improve the positioning accuracy of the system and reduce the cost of the system is crucial to the real applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.
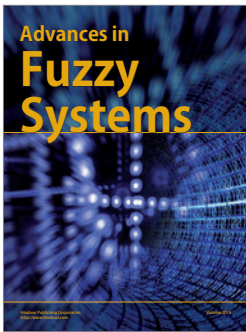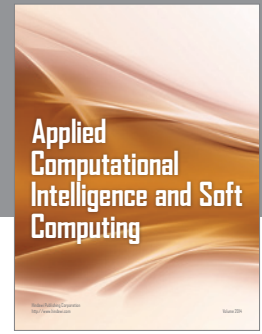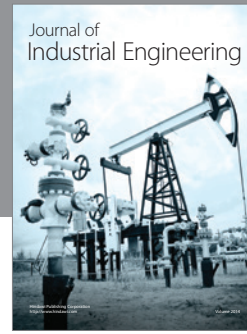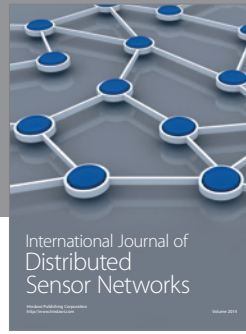
Acknowledgments

References

[1] Y. Wang, J. Zhao, and T. Fukushima, "LOCK: a highly accurate, easy-to-use location-based access control system," in *Location and Context Awareness: 4th International Symposium, LoCA 2009 Tokyo, Japan, May 7-8, 2009 Proceedings*, vol. 5561 of

Lecture Notes in Computer Science, pp. 254–270, Springer, Berlin, Germany, 2009.

[2] V. W. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang, "Collaborativex filtering meets mobile recommendation: a user-centered approach," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI '10)*, pp. 236–241, Atlanta, Ga, USA, July 2010.

[3] S. Kizhakkepurayil, O. Jeon-Yeoul, and Y. Lee, "Mobile application for healthcare system—location based," in *Innovations and Advances in Computer Sciences and Engineering*, T. Sobh, Ed., pp. 297–302, Springer, Dordrecht, The Netherlands, 2010.

[4] E. Kaplan and C. Hegarty, Eds., *Understanding GPS Principles and Applications*, Artech House, 2nd edition, 2005.

[5] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, IEEE, Tel Aviv, Israel, March 2000.

[6] E. S. Bhasker, S. W. Brown, and W. G. Griswold, "Employing user feedback for fast, accurate, low-maintenance geolocationing," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom '04)*, pp. 111–120, March 2004.

[7] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On indoor position location with wireless LANs," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, pp. 720–724, September 2002.

[8] P. Kontkanen, P. Myllymaki, T. Roos, H. Tirri, K. Valtonen, and H. Wettig, "Topics in probabilistic location estimation in wireless networks," in *Proceedings of the 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04)*, vol. 2, pp. 1052–1056, IEEE, September 2004.

[9] H. Mehmood, N. K. Tripathi, and T. Tipdecho, "Indoor positioning system using artificial neural network," *Journal of Computer Science*, vol. 6, no. 10, pp. 1219–1225, 2010.

[10] J. Yim, "Introducing a decision tree-based indoor positioning technique," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1296–1302, 2008.

[11] V. Moghtadaiee and A. G. Dempstera, "Indoor location fingerprinting using FM radio signals," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 336–346, 2014.

[12] M. Oussalah, M. Alakhras, and M. I. Hussein, "Multivariable fuzzy inference system for fingerprinting indoor localization," *Fuzzy Sets and Systems*, vol. 269, pp. 65–89, 2015.

[13] J. Talvitie, M. Renfors, and E. S. Lohan, "Distance-based interpolation and extrapolation methods for RSS-based localization with indoor wireless signals," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1340–1353, 2015.

[14] A. Smola, "Regression estimation with support vector learning machines," Tech. Rep., Technische Universität München, 1996.

[15] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT2 Technical Report Series NC2-TR-1998-030, 1998.

[16] V. S. Feng, T. C. Wang, S. Y. Chang, and H.-P. Ma, "Location estimation in indoor wireless networks by hierarchical support vector machines with fast learning algorithm," in *Proceedings of the International Conference on System Science and Engineering (ICSSE '10)*, pp. 321–326, July 2010.

[17] R. Battiti, M. Brunato, and A. Villani, "Statistical learning theory for location fingerprinting in wireless LANs," Tech. Rep. DIT-02-0086, Università di Trento, Trento, Italy, 2002.

[18] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825–845, 2005.

[19] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 134–152, 2005.

[20] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 981–994, 2008.

[21] Z.-L. Wu, C.-H. Li, J. K.-Y. Ng, and K. R. P. H. Leung, "Location estimation via support vector regression," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 311–321, 2007.

[22] Z. A. Deng, Y. B. Xu, and D. Wu, "Indoor positioning in WLAN environment based on support vector regression and space partitioning," *Advanced Materials Research*, vol. 204–210, pp. 1599–1602, 2011.

[23] M. H. Vahidnia, M. R. Malek, N. Mohammadi, and A. A. Alesheikh, "A hierarchical signal-space partitioning technique for indoor positioning with WLAN to support location-awareness in mobile map services," *Wireless Personal Communications*, vol. 69, no. 2, pp. 689–719, 2013.

[24] Z.-A. Deng, Y.-B. Xu, and L. Ma, "Indoor positioning via nonlinear discriminative feature extraction in wireless local area network," *Computer Communications*, vol. 35, no. 6, pp. 738–747, 2012.

[25] A. Smola and B. Schölkopf, "A tutorial on support vector regression," Neuro COLT Technical Report TR-1998-030, Royal Holloway College, London, UK, 1998.

[26] B. Sklar, "Rayleigh fading channels in mobile digital communication systems. I. Characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, 1997.

[27] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLAN's received signal strength indication for indoor location fingerprinting," *Pervasive and Mobile Computing*, vol. 8, no. 2, pp. 292–316, 2012.

[28] T. King, T. Haenselmann, and W. Effelsberg, "On-demand fingerprint selection for 802.11-based positioning systems," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '08)*, pp. 1–8, IEEE, Newport Beach, Calif, USA, June 2008.