Ensemble Strategy Utilizing a Broad Learning System for Indoor Fingerprint Localization

Chen Wu, Tie Qiu, *Senior Member, IEEE*, Chaokun Zhang, *Member, IEEE*, Wenyu Qu, and Dapeng Oliver Wu, *Fellow, IEEE*

***Abstract*—Indoor positioning technology based on Wi-Fi fingerprint recognition has been widely studied owing to the pervasiveness of hardware facilities and the ease of implementation of software technology. However, the similarity-based method is not sufficiently accurate, whereas the offline training of the neural network-based method is overly time consuming. An efficient model with high positioning accuracy is therefore not yet available. We propose a stacking ensemble broad learning localization system using channel state information as a fingerprint, which is termed EnsemLoca. A bootstrapping method is used to build the training set, which enables the EnsemLoca system to build the base learner in parallel by using bagging. The broad learning system (BLS), which is a novel neural network model, as a base learner, not only has the advantage of time complexity but also offers a sparse representation in which the features are filtered. A unique base learner is constructed by randomly selecting the samples and features, and they are combined by stack generalization. The experimental results show that the EnsemLoca system achieves higher accuracy than several machine-learning algorithms in both line-of-sight (LOS) and non-LOS environments, and is even stronger than deep neural networks characterized by accuracy. At the same time, it has the same theoretical complexity as BLS, which greatly reduces the offline training time.**

***Index Terms*—Broad learning system (BLS), channel state information (CSI), intelligent localization, Internet of Things.**

I. Introduction

With the emergence of next-generation 5G wireless communication systems and the continuous development of smart cities, the construction of intelligent Internet-of-Things systems based on indoor positioning has become increasingly important. Generally, the global navigation satellite system (GNSS) with a positioning accuracy of several meters has solved most outdoor positioning problems and is widely used in various commercial and military scenarios [1]. However, the signal propagation model of a GNSS system is only effective in a line-of-sight (LOS) environment. The non-LOS (NLOS) indoor environment is characterized by multipath effects, fading, and delay, which means a new method needs to be developed to build an indoor positioning system [2]. Because of the framework and characteristics of mobile devices, Wi-Fi fingerprint-based location systems are attracting increased attention from researchers.

The Wi-Fi-fingerprint-based location algorithm can be divided into two stages: 1) offline training and 2) online matching [3]. In the offline phase, the algorithm uses an existing fingerprint data set, i.e., a data set that was built before to learn the relationship between the fingerprint and position. In the online matching stage, the most likely position of the fingerprint is obtained by contrasting the similarity. One of the advantages of this method is that it is not necessary to know the position of the AP and the transmission attenuation model of the signal.

Because of the simplicity and easy access to the received signal strength indication (RSSI), many existing positioning systems use RSSI as fingerprints. For example, the RADAR system [4] uses a K-nearest neighbor (KNN) algorithm to calculate the similarity between the RSSI, and Horus [5] uses a Gaussian regression method to obtain the most possible RSSI. However, the information provided by the RSSI fingerprint itself is limited and coarse grained, which makes it difficult to reflect the relationship between the position and the signal. Channel state information (CSI) signals could be effectively used as an alternative to obtaining fine-grained and informative fingerprints and the use of CSI signals has been confirmed to improve the accuracy of positioning systems [6], [7].

The KNN-based system [8] does not need training, while it needs to match all fingerprints in the online stage. The increase of offline fingerprint databases is likely to make the complexity of the system difficult to accept. Although the CSI is used, the low efficiency of the feature selection method made the effect not ideal. In contrast, the neural network method does not need to compare all the fingerprint data and only needs to train the weight of the model by using a back-propagation algorithm in the offline stage. The complexity of this method depends on the number of neurons rather than the size of the scenarios. Although the deep neural network (DNN) model can achieve good positioning results, this method generally requires too much training time in the offline

Manuscript received April 25, 2021; revised June 16, 2021; accepted June 29, 2021. Date of publication July 15, 2021; date of current version February 4, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1703601; in part by the Joint Funds of the National Natural Science Foundation of China under Grant U2001204; in part by the Tianjin Science Foundation for Distinguished Young Scholars under Grant 20JCJQJC00250; in part by the China Postdoctoral Science Foundation under Grant 2020M680906; and in part by the Hebei Province High-Level Talent Funding Project under Grant B202003027. *(Corresponding authors: Tie Qiu; Chaokun Zhang.)*

Chen Wu, Tie Qiu, Chaokun Zhang, and Wenyu Qu are with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin Key Laboratory of Advanced Networking, Tianjin University, Tianjin 300350, China (e-mail: wing@tju.edu.cn; qiutie@ieee.org; zhangchaokun@tju.edu.cn; wenyu.qu@tju.edu.cn).

Dapeng Oliver Wu is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: wu@ece.ufl.edu).

Digital Object Identifier 10.1109/JIOT.2021.3097511

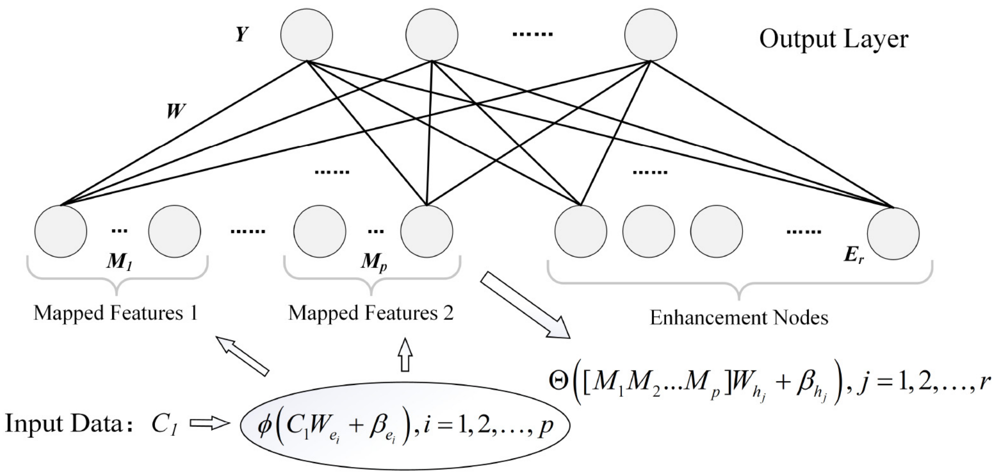

Y
Output Layer
W
M_1
M_p
E_r
Mapped Features 1
Mapped Features 2
Enhancement Nodes
Input Data：C_1
$\phi\left(C_1W_{e_i}+\beta_{e_i}\right), i=1,2,\ldots,p$
$\Theta\left([M_1M_2\ldots M_p]W_{h_j}+\beta_{h_j}\right), j=1,2,\ldots,r$

Fig. 1. Architecture of BLS based on CSI.

phase [9]. The large number of hyperparameters complicates offline training and extends the training time [1]. Besides, in the actual large-scale scene, the marginal benefit of improving the accuracy of the decimeter level has been far less than the marginal cost.

To summarize, the above-mentioned algorithms are problematic in three different ways. First, based on methods that involve a similarity comparison, such as KNN, the time consumed in the online location is proportional to the size of the fingerprint database, which cannot be applied to large scenes. Second, the DNN model has many hyperparameters to choose from and uses backpropagation, which considerably increases the time complexity of the training phase. Third, effective data noise processing and feature selection methods to improve the positioning accuracy of CSI do not exist. We propose an EnsemLoca system using the CSI signal based on the broad learning system (BLS) and ensemble learning to overcome these problems. The main contributions of this study are as follows.

1) We propose the EnsemLoca algorithm, which is based on BLS. The complexity of BLS in online computation is only related to the weight of the model, which can solve the problem of online time consumption. At the same time, the use of ridge regression allows BLS to be quickly trained even in the offline phase.
2) We combine the bagging framework with BLS. This combination makes parallel training possible and maintains the speed of BLS. In addition, to improve the accuracy, we propose a method with the ability to construct a BLS with unique recognition. Samples and features are randomly selected in different ways, and sparse representation is used to build distinctive recognition, which deepens the recognition of EnsemLoca so that it can learn hidden features.
3) We compared EnsemLoca with many existing machine-learning algorithms, such as DNN, KNN, and Horus. The results verified that EnsemLoca is more accurate and offers a time advantage in both NLOS and LOS environments. Furthermore, we also explored the influence of the properties of EnsemLoca on the final results.

The remainder of this article is organized as follows. Section II presents basic information about EnsemLoca. Section III describes the various implementations within the EnsemLoca system in detail. Section IV presents the simulation and evaluation of the EnsemLoca from various aspects. Section V reviews related work and Section VI summarizes this article.

II. Preliminaries

A. CSI

CSI signals contain highly important and practical channel data in wireless communication, and represent the propagation characteristics of communication links. Unlike the RSSI, which only describes the signal strength, CSI describes the joint effects of scattering, fading, and power attenuation of each subcarrier in the channel under the influence of the environment. However, before the Intel 5300 NIC appeared, it was difficult to obtain the CSI that belongs to the physical-layer information. The Intel 5300 NIC implements an OFDM system with 56 subcarriers, and the information of 30 subcarriers can be obtained by using the Linux 802.11n CSI tool [10]. The common multiple-input–multiple-output (MIMO) linear model can be expressed as follows:

$$\vec{Y} = H \cdot \vec{X} + \vec{N} \tag{1}$$

where $\vec{Y}$ is the received signal vector, H is the CSI matrix, and $\vec{N}$ is the white Gaussian noise. For each subcarrier, the CSI signal is a complex value, which can be expressed as follows according to the Euler formula:

$$H = |H|e^{i\angle H} \tag{2}$$

where $\angle H$ is the phase and $|H|$ represents the amplitude. The amplitude reflects the characteristics of the channel in the frequency domain and shows great stability with respect to the changes in the given location with time. Therefore, the amplitude of 30 subcarriers of the CSI is selected as a feature to train the BLS. Since three antennas are used, 90 subcarriers are obtained for each position as the characteristics of

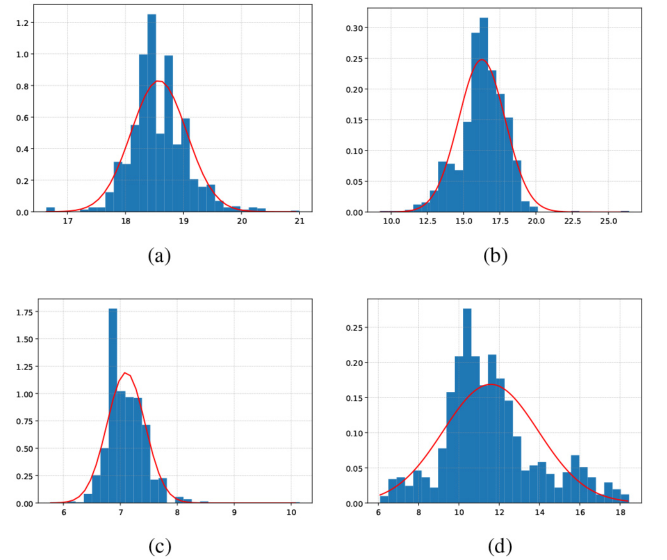


Fig. 2. Frequency of amplitude in different features and locations. (a) SpotA_feature1. (b) SpotB_feature1. (c) SpotA_feature2. (d) SpotB_feature2.

each sample. The processing of CSI and a further detailed description is provided in Section III-B.

B. BLS

The development of the DNN model has resulted in its nonlinear expression ability attracting increasing attention. To date, many researchers have applied DNNs to indoor positioning fields and have achieved good results. However, owing to the excessive number of hyperparameters and the lack of theoretical guidance of the model architecture, a DNN requires much time to optimize in the offline phase [11]. Consequently, BLS, a neural network model with only one layer of the horizontal architecture, was proposed as an improvement to deep learning [12]. With fewer hyperparameters and no backpropagation, BLS offers fast training and easier optimization, even if the precision is degraded.

The architecture of the BLS is shown in Fig. 1. Mapped feature nodes are generated by function mapping with randomly generated W_{e_i} and β_{e_i}, which satisfy the normal distribution. Then, the enhanced nodes are obtained by using randomly generated orthogonal weights and activation functions. Finally, feature nodes and enhancement nodes are directly connected to the input layer, and ridge regression is used to calculate the pseudo-inverse to obtain the weight, which significantly shortens the processing time. This prompted us to use BLS to reduce the time and complexity of the training phase, which is described in Section III-C.

C. Bagging

Bagging, one of the most intuitive and classic frameworks in ensemble learning, expresses the idea that the value of 1 and 1 equals 3. For data-sensitive algorithms, such as neural networks, by constructing different training sets and training methods, this framework can improve the overall accuracy by integrating multiple algorithms [13]. BLS is a neural network, which means that it is also suitable for the use in combination with this framework.

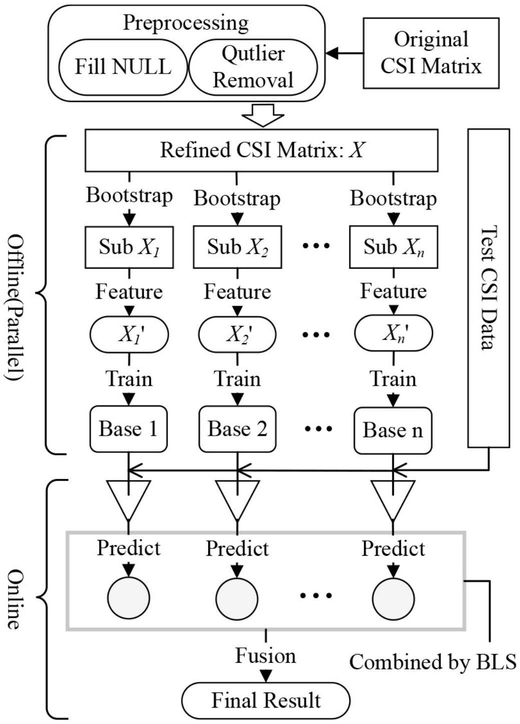


Fig. 3. Architecture of the EnsemLoca system.

Next, the bagging framework is briefly introduced. For a training set with m samples

$$L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \tag{3}$$

where x_i is a vector and y_i is a label, we can obtain a predictor $\varphi(x, L)$. By m times random sampling with replacement (bootstrap), a new training set L' can be constructed, which has the same probability distribution and samples as the original training set. Therefore, the final result $\varphi_A(x)$ can be expressed as follows:

$$\varphi_A(x) = \sum_{1}^{n} w_i \varphi(x, L_i) \tag{4}$$

where w_i is the weight of different predictors. In addition, in this study, we do not simply apply the bagging framework directly, but modify it to make it more suitable for application scenarios. The detailed organizational structure is described in Section III-D.

III. System Design of EnsemLoca

A. Architecture Description

Fig. 3 shows the architecture of the EnsemLoca system. The architecture is divided into data preprocessing, an offline training phase, and an online localization phase. In the data preprocessing phase, the data are refined in terms of filling in null values and removing outliers. Unlike other studies, we do not rerepresent the data in the preprocessing phase, but rather process the data using the BLS, where sparse representation is a good way to extract the data features.

Algorithm 1: Data Preprocessing

```
Input: raw CSI matrix $H$ with the size of $3 \times 30$ (single
       packet), number of sample positions $N_l$, sampled
       packets per position $s$, NULL mark array $f$;
Output: refined CSI matrix $C_1$
Initialize $C_1, f = \emptyset$
for $i = 1{:}s * N_l$ do
    acquire next $H$;
    $H$ = abs ($H$) //greater than or equal to 0;
    if $H = 0$ then
        $H$ = null;
        append $i$ to $f$ //record position for filling;
    end
    append $H$ to $C_1$;
end
foreach $column \in C_1$ do
    $\mu$ = Mean ($column$) //average of the column;
    $\sigma$ = Std ($column$) //standard deviation;
    fill null by $(\mu, \sigma, f)$;
    foreach $sample \in column$ do
        if $sample - \mu \geq 3\sigma$ then
            update $sample$ by $\mu$;
        end
        update $sample$ by equation (5)
    end
end
```

Considering the speed of BLS in training and testing, it is used as a basic learner. This is followed by the offline phase, in which the refined data are used to construct n copies of the training set of the same distribution by bootstrap and feature selection. A bagging framework is used to train and combine these n base learners. Because the bagging framework can be trained in parallel, the complexity is only related to the complexity of a single BLS, and the time required for training does not increase with the number of predictors, which maximizes the advantage of BLS as a base learner in terms of training speed. Then, the obtained n predictions are sent to the next layer to train the weights of the combiner, which means the completion of the training phase.

In the online localization phase, the signal to be tested is fed into the base learner, and the test signal is reconstructed by using the feature selection pattern saved in EnsemLoca. Then, n predictions are derived, and the combiner is fused to obtain the final localization result. In the next section, we describe EnsemLoca in detail in terms of data preprocessing, base learners, and sampling and combination strategies.

B. Data Preprocessing

Before sending data into the EnsemLoca system, it is necessary to ensure its integrity and correctness, considering that the data set contains null and error values because of collection and hardware errors. In both cases, we replace the missing value with the average of attributes. A statistical method is applied to find inappropriate values. Assuming a Gaussian normal distribution $X \sim N(\mu, \sigma^2)$, the probability of data lying in the interval $[\mu - 3\sigma, \mu + 3\sigma]$ is 99.74%. The 3σ is used to ensure that more normal fluctuation data are included, while values beyond this interval are considered probabilistic undesirable. However, before we provide a detailed explanation of the preprocessing stage, we need to explain the rationality of this processing.

To test this hypothesis, multiple attributes of 2000 packets at each location are tested. In Fig. 2, the horizontal and vertical coordinates are the amplitude and frequency, respectively, and the result is consistent with our assumed Gaussian normal distribution. Furthermore, we choose two positions: A and B, and two features: 1) feature 1 and 2) feature 2, to compare their performance under the influence of the environment. The amplitude $|H|$ in formula (2) is an unstable value as a feature just like RSSI. However, the RSSI signal has only one attribute, while $|H|$ has 90 attributes to describe the channel in a fine-grained way. The results show that different attributes of the same position are affected by the environment to a different degree. Meanwhile, using stable CSI instead of unstable RSSI is the consensus of researchers [6]–[8]. Therefore, we can choose the stable attribute of the CSI signal to further improve the accuracy of the system.

The aggregated form of the data and the pseudocode of the process are shown in Algorithm 1. The algorithm processes all the data iteratively, converts it to amplitude and marks the null position. After filling in the missing and error values, we standardize the data to eliminate the influence of different attribute dimensions, which can improve the speed and accuracy of training. Data standardization uses the Z-score method as

$$Z = \frac{X - \mu}{\sigma} \tag{5}$$

where X represents the data of subcarriers, μ is the mean value, and σ represents the standard deviation of the corresponding subcarriers. In this way, data that are preliminarily processed are obtained, and because the features are reconstructed using a sparse representation in BLS, we do not transform the raw data using principal component analysis (PCA) and other filtering techniques.

C. Base Learner: BLS

In this section, we introduce the internal implementation of BLS as the base learner in detail, and its pseudocode is shown in Algorithm 2. First, to obtain the bias terms, we obtain $C_2 \in R^{N \times (F+1)}$ by appending one column to C_1' (unique training set), each of which is 0.1. Then, the characteristic is extracted from the reconstructed data C_2 to build the mapped feature node. All mapped feature nodes are divided into p groups from M_1 to M_p, with each group having q nodes. In general, M_i can be regarded by nonspecific mapping function ϕ as

$$\phi\big(C_1 W_{e_i} + \beta_{e_i}\big), \quad i = 1, 2, \ldots, p \tag{6}$$

where W_{e_i} is a random weight matrix and β_{e_i} is the white noise. As mentioned above, the additional column of C_2 is added to simulate white noise. For group M_i, matrix A_i is defined as $A_i = C_2 * W_{e_i}$, of which the size is $(F + 1) \times q$.

Algorithm 2: EnsemLoca System

```
Input: Refined CSI matrix: $C_1$, training label: $Y$, number
       of base learners: $n$, selected samples and features:
       $S_s$ and $S_f$, hyper-parameters of BLS: $p, q, r$,
       testing data: $D_t$
Output: Final prediction $F_p$
//offline training phase;
for $k = 1 : n$ do
    //build unique training set;
    Obtain $C_1'$ by $C_1, S_{s_k}$ and $S_{f_k}$;
    for $i = 1 : p$ do
        Obtain $C_2$ by adding a column to $C_1'$;
        Generate random weight matrix: $W_{e_i}$;
        $A_i = C_2 * W_{e_i}$ and generate $W_i$ by (7);
        $M_i = C_2 * W_i$ //mapped feature node
    end
    $M = [M_1, \ldots, M_p]$, then orthogonal normalized;
    Obtain $M'$ by adding a column to $M$;
    $E = tanh(M' * W_{h_j})$ //the same with $W_{e_i}$;
    $W_k = [M|E]^+ Y$ //weight of the k-th BLS;
    Get k-th prediction $F_k$;
end
//acquire weight between base learner and combinator;
$W_{stack} = \mathrm{BLS}(F_1, \ldots F_n, Y)$;
//online testing phase;
for $k = 1 : n$ do
    Construct input data by $S_{s_k}$, $S_{f_k}$ and $D_t$;
    Get k-th prediction $F_k$;
end
$F_p = \mathrm{BLS}(F_1, \ldots F_n, W_{stack})$;
```

Normalized $\hat{A}_i$ is used to solve the sparse matrix W_i as

$$\arg\min_{\hat{W}_i} : \left\| \hat{A}_i \hat{W}_i - C_2 \right\|_2^2 + \lambda \left\| \hat{W}_i \right\|_1 \tag{7}$$

where λ is sparse coefficient. We denote the mapped feature node $M = [M_1, \ldots M_p]$, where $M_i = C_2 * W_i$. Furthermore, the enhanced nodes $E = [E_1, \ldots, E_r]$ with nonlinear expression ability are constructed by M, where E_j can be expressed as

$$\Theta\left(M_j W_{h_j} + \beta_{h_j}\right), \quad j = 1, 2, \ldots, r. \tag{8}$$

Different from the mapped feature node, orthonormal M makes the network more expressive by mapping to a high-dimensional subspace, whereas β_{h_j} has the same meaning as β_{e_i}. In this study, we use tanh as the activation function Θ given by

$$y = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{9}$$

Incidentally, enhanced nodes can choose different activation functions, such as sigmod and ReLU, for E_j and E_k when $j \neq k$. Finally, the output layer Y can be described as

$$Y = [M|E] * W \tag{10}$$

where W is the overall weights between the input feature and output label. Ridge regression with regularization coefficient γ is used to rapidly calculate W, which means that BLS training is complete.

D. Sample and Combination Strategy

After preprocessing, we have the reconstructed CSI matrix C_1, but it can only train one predictor. We sample the data set with replacement (bootstrap) to build different training sets C_1', which have the same distribution for training predictors as base learners. Instead of building subsets of the same size as the original training set, the subsets X_i are constructed by the threshold α $(0 < \alpha \leq 1)$. The probability of an unselected sample can be expressed as

$$\lim_{m \to +\infty} \left(1 - \frac{1}{m}\right)^{\lfloor \alpha m \rfloor} \approx \left(\frac{1}{e}\right)^{\alpha} \tag{11}$$

where m is the number of samples and $\lfloor \alpha m \rfloor$ represents the sampling size. Obviously, the constructed training set decreases as α decreases. Researchers always assume that the larger the sample, the more effective the learning [14], [15]. However, based on Experiment E, reducing the sample size within a certain range does not adversely affect the accuracy in an obvious way. In addition, we consider reduction to effectively reduce the computational complexity and extract more fine-grained information.

For the same reason, β is used to randomly sample features. Sampling without replacement is used here to acquire more comprehensive information. Then, the identifiers of selected features for each base learner are recorded and used in the online positioning stage to build a consistent relationship between the features of a sample and the base learner. Although the reduction of the training features lowers the accuracy of the individual BLS, it adds diverse prediction perspectives to the entire system, laying the foundation for the subsequent success of the combined strategy.

Many options are available for combinators, linear regression, logistic regression, and neural networks. In this study, we chose BLS, a lightweight neural network model as a combinator to increase the speed and accuracy. Nonlinear mapping allows different base learners to contribute differently to the same point and allows the same base learner to contribute differently to each point.

The theoretical complexity of EnsemLoca is only the sum of two BLSs, which greatly speeds up the training process compared to DNN. In addition, the structure of EnsemLoca also reflects our understanding of the philosophy of life: win–win cooperation, where we apply the modern idea of division of labor. This ideology allows each division to be deeper, and cooperation further combines the strengths of each division.

IV. Experiments and Analysis

A. Experimental Environment

We used a Dell XPS desktop computer as the transmitter and a Dell laptop as the receiver to obtain the data set. Each computer was equipped with an Intel 5300 NIC and had 64-bit Ubuntu 12.04 LTS installed. Raw CSI data are received by using the Linux 802.11n CSI Tool with a modified NIC driver. Data were collected from two selected rooms to build a data

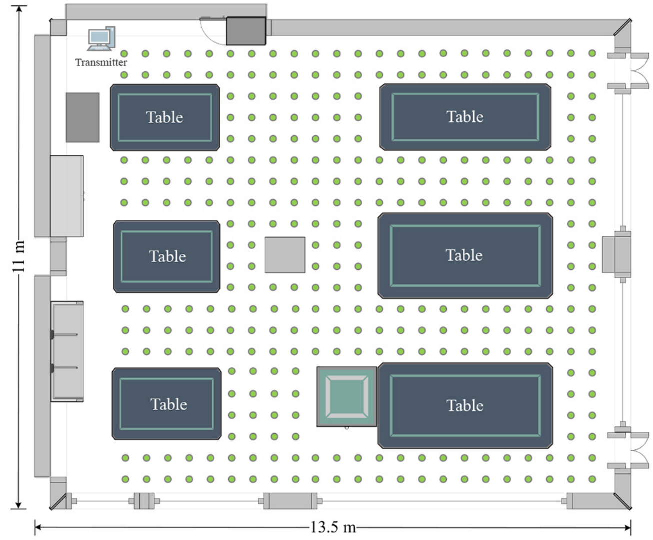


Fig. 4. Layout of laboratory.

set to test the performance of EnsemLoca in NLOS and LOS scenarios. We take approximately 5 s to ensure that each point receives more than 1000 packets through the monitoring mode, where the sending parameters can be set arbitrarily to improve the efficiency.

1) A laboratory in building No. 47 of Tianjin University. The laboratory contains many tables, chairs, and tools, such as types of sensors, and covers an area of 11 × 13.5 m^2. As shown in Fig. 4, the transmitter is placed in the upper left corner, and the gray block in the center is the load-bearing wall. There are 317 sampling spots, separated by a distance of 50 cm, which are the positions at which we collect signals. During the collection process, peoples' movements were not restricted, neither were the mobile phones, computers, and Wi-Fi signals removed from the room, which was used to simulate an NLOS scene in practical applications.
2) A conference room in building No. 55 of Tianjin university, which contains an open environment on the left and tables on the right and covers an area of 10.5 × 12 m^2. As shown in Fig. 5, the transmitter is also placed at the top left, and the laptop is also placed on the ground for signal collection. Unlike the laboratory that was used to simulate NLOS scenarios, where signals were collected at 176 points with an interval of 60 cm, the movement of personnel is limited to reduce interference with the aim of simulating LOS scenarios.

In this study, root mean square error (RMSE) is used to measure the error between the predicted value and actual value. For the m test cases, the RMSE can be expressed as

$$\mathrm{RMSE}(C_1, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x_i) - y_i)^2} \tag{12}$$

where $h(x_i)$ denotes the predicted location and y_i is the real location. Although the RMSE can be used to measure the overall prediction, we hope to additionally obtain the distribution. Therefore, the cumulative distribution function (CDF) diagram is used to show the internal structure. In addition, stratified fivefold cross validation is used to avoid contingency, in which 80% of the data is used for training and 20% of the data is used for testing. The remaining parameters of the EnsemLoca mechanism are listed in Table I. For comparison purposes, the same parameters are set in the two scenarios.

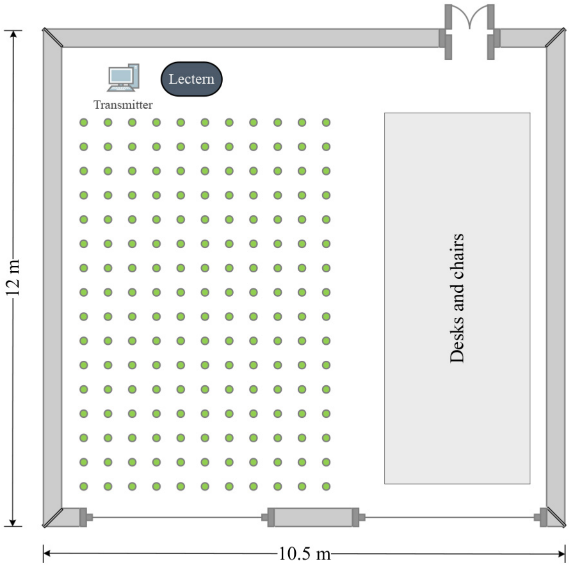


Fig. 5. Layout of meeting room.

TABLE I
PARAMETERS OF ENSEMLOCA

Parameters	Lab & Meeting Room
Feature nodes in base learner q	20
Feature groups in base learner p	2
Feature nodes in fusion phase q'	10
Feature groups in fusion phase p'	2
Enhancement nodes r	100
Regularization coefficient γ	2^{-30}
Sparse coefficient λ	0.001
Number of base learner n	20
Threshold of sample α	0.3
Threshold of feature β	0.3
Number of packets per position	50

B. Overall Performance

It is necessary to verify the efficiency of EnsemLoca in NLOS and LOS environments. RADAR uses KNN to match fingerprints, in which its K is briefly set to 5 to balance the matching error. The parameters of DNN are listed in Table II, in which the dropout layer is set to prevent overfitting. First, we discuss the complexity of each algorithm in the training phase. KNN is well known for not requiring training, compared with Horus, which uses posterior distribution, and DNN which uses backpropagation with a large amount of training time. The training time of each algorithm is listed in Table III, and BLS is the fastest one except for KNN. SWIM [16] searches for the hyperplane in the training phase, which its time consumed is similar to DNN. The time of Horus is longer

TABLE II
PARAMETERS OF DNN

Parameters	Laboratory & Meeting Room
Input layer	90
Fully connected layer	100
Activation function	*Sigmoid*
Dropout layer	0.3
Output layer	2
Optimizer	*Adam*
Batch size	32
Epochs	100

than that of BLS because it needs to fit the distribution, but it is still far less than the training time of DNN. What excites us is that the training complexity of BLS is even two orders of magnitude lower than that of DNN. Furthermore, the complexity of EnsemLoca can be seen as the sum of the two BLS, but the cost time is much longer than we expected. This is because the global interpreter lock (GIL) of Python prevents multiple native threads from viewing Python bytecodes at once in order to ensure thread safety, which means that EnsemLoca can be implemented by other languages to solve this problem and achieve the expected time consumption.

Next, we consider the complexity in the online phase. The time the model requires for online location based on similar matching, such as RADAR and Horus, increases linearly with the width of the scenarios, in which its theoretical complexity can expressed as O$(k * N)$, where N stands for the size of fingerprint database and k stands for the unfixed constant. Although some existing works [17]–[19] have achieved high accuracy, the complexity of the online phase also has the same problem because of the use of KNN in the system. The complexity of SWIM can be expressed as $O(N_s * N_p)$, where N_s is the number of support vectors and N_p is the number of positions in the scene. Although it is not necessary to compare with each fingerprint, the two parameters will increase with the increase of samples and scenes, which means that the algorithm is better than KNN, but it is still related to the scene size.

In contrast, the online complexity of the neural network algorithm is invariable for a fixed network structure. Ignoring other effects such as activation function, the online theoretical complexity of DNN can be expressed by the number of calculations between neurons. It can be expressed as O$(D_i * D_f * D_o)$, where D_i, D_o, and D_f are the dimensions of input, dimensions of output, and numbers of neurons. Similarly, the theoretical complexity of BLS and EnsemLoca can be expressed as O$(D_i * (p * q + r) * D_o)$, in which $(p * q + r)$ stands for the number of neurons and has the same scale with D_f.

The positioning accuracy is shown in Table III. As expected, the NLOS error is larger than the LOS error, and DNN and Horus are more accurate than KNN. Although the accuracy of SWIM in LOS is similar to that of DNN, it is greatly reduced in NLOS. In NLOS, the performance of BLS is similar to that of Horus, but the time cost is one order of magnitude lower.

TABLE III
OVERALL ACCURACY

Algorithms	Laboratory		Meeting Room	
	Mean(m)	Train time(s)	Mean(m)	Train time(s)
EnsemLoca	**2.5378**	8.0979	**1.1497**	4.4056
BLS [12]	3.1969	0.2795	1.5139	0.0883
SWIM [16]	3.1687	30.7275	1.4229	9.4526
DNN	2.9002	36.4992	1.4118	18.7743
HORUS [5]	2.9968	9.2688	1.8532	2.2588
RADAR [4]	3.3905	None	1.9713	None

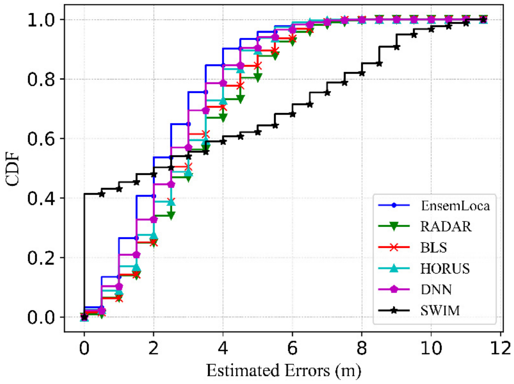


Fig. 6. CDF of mean errors in laboratory.

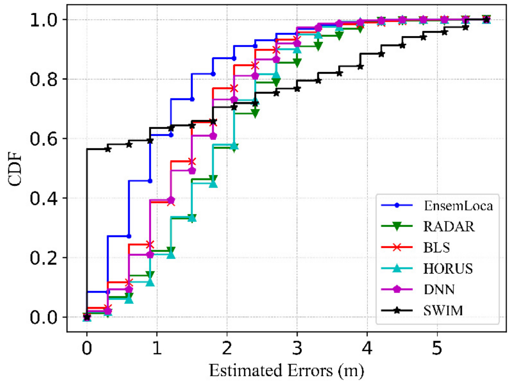


Fig. 7. CDF of mean errors in meeting room.

However, in LOS, the accuracy of BLS is much higher than that of HORUS and RADAR. This result confirms our judgment of BLS, i.e., that it is an efficient and potentially accurate predictor, and enables us to confidently build a BLS-based framework. Most surprisingly, EnsemLoca has the same level of complexity as BLS, but its accuracy is improved by nearly 13% and 23%, respectively, compared to DNN, which is characterized by precision. Figs. 6 and 7 show the distribution of the prediction results. SWIM is very accurate in predicting points with differences. However, when some similar points appear, the performance will decline sharply. The neural network model is more dominant in LOS, whereas EnsemLoca delivers superior performance in both scenarios.

In order to verify the robustness of our algorithm, comparative experiments are carried out based on the data set of Widar2.0 [20]. The classroom is a rectangle of 6 × 5 m^2,

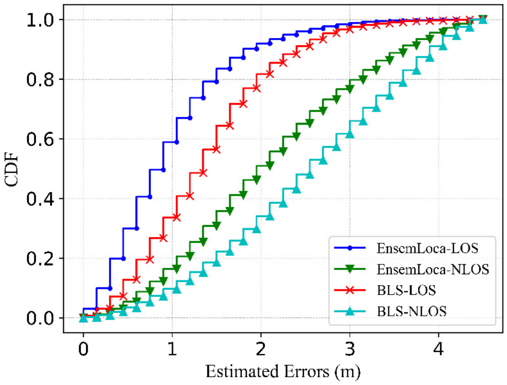


Fig. 8. Performance of EnsemLoca and BLS.

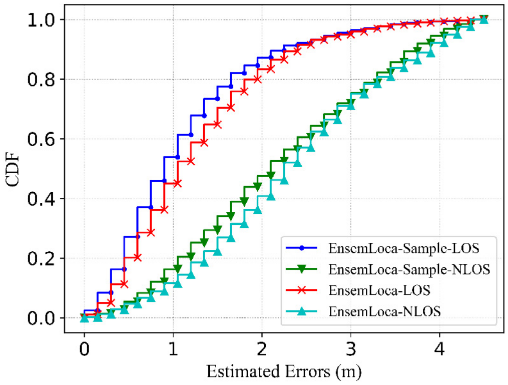


Fig. 9. Performance of sampled and unsampled methods.

and the experimental results are shown in Table IV. As we expected, EnsemLoca has the highest accuracy, and the SWIM is not appeared because of the lack of ability of regression problem. Horus gets the lowest standard deviation by fitting training data, which leads to the decrease of accuracy. In contrast, RADAR has achieved good performance. The online time of RADAR in large scenes is unacceptable, but it is still a good choice for small scenes with multipoint data collection. Although we have demonstrated the superiority of EnsemLoca in terms of time and precision, it is also necessary to discuss the effects of various parameters and structures in EnsemLoca by conducting further experiments.

TABLE IV
ACCURACY IN CLASSROOM OF WIDAR2.0

Algorithms	Classroom		
	Mean (m)	Std (m)	Train time (s)
EnsemLoca	**1.4752**	0.7304	0.5765
BLS [12]	1.7031	0.7729	0.0608
DNN	1.6623	0.5781	9.2121
HORUS [5]	1.7449	0.5433	1.3753
RADAR [4]	1.6472	1.1791	None

C. Impact of Ensemble

As shown in Fig. 8, the integrated approach of EnsemLoca and that of the nonintegrated BLS are placed in a CDF diagram. Regardless of effectiveness, the convergence rate of EnsemLoca is always higher for LOS than NLOS, which is attributable to the characteristics of the data. In addition, EnsemLoca converges more quickly for the same scene with improvements of 0.65 and 0.36 m, respectively.

This does not suggest that the capability of EnsemLoca in LOS scenarios is limited. When the accuracy of the improvement is expressed as a percentage, the accuracy for the LOS scenario is 3% higher than that for NLOS. This is because the accuracy of positioning algorithms in LOS scenarios is generally higher. The same can be concluded from the curve of EnsemLoca-LOS, which increases sharply in CDF, indicating that the ensemble is helpful for improving the accuracy. For data preprocessing, we also do a comparative experiment. However, because of the few wrong data collected, there is no obvious difference between the experimental results.

To further decompose the integration method used in EnsemLoca, two points are worth considering, namely, "the way in which to combine the predictors" and "the types of predictors to combine." For the former, we initially used simple and fast linear regression, but the performance improvement was not obvious. The use of linear fitting means the tolerance of overall performance of BLS, which consists of its strengths and weaknesses. In other words, the contribution of base learner affects every position in the same way, which is contrary to our goal that each BLS uses its expertise to its advantage in certain positions. Fig. 8 shows that each BLS can only focus on the information of a position with his unique opinion, giving full play to the advantages of division and cooperation through a nonlinear mapping relationship. Another key point that must be noted is that EnsemLoca uses the sampling method to build a base learner, which relates to the question "the types of predictors to combine."

D. Impact of Sampling

In this section, we discuss the way in which the sampling impacts the performance of the proposed EnsemLoca. As shown in Fig. 9 and Table V, it is obvious that the sampling method can greatly improve the accuracy. The thresholds of samples and features of EnsemLoca-noSample for comparison are set to 1.0, and the other parameters are consistent with those of EnsemLoca. Without sampling, EnsemLoca is slightly worse than DNN in NLOS scenarios and more accurate than DNN in LOS scenarios and the overall accuracy is similar to that of DNN. Moreover, comparison of the Std shows that the sampling method also reduces the deviation of data. The diversity of the base learners can be increased by sampling, thus expanding the recognition range and overcoming the extent to which data values contradict each other.

The range of sampling also affects the accuracy of our preliminary experiments. Low and high sampling rates are likely to have a large impact on the accuracy; the latter is expected to marginally improve the accuracy, whereas the former may lead to incomplete learning because of an insufficient number of samples. However, we find that the features learned by the high sampling rate base learner are more moderate, which

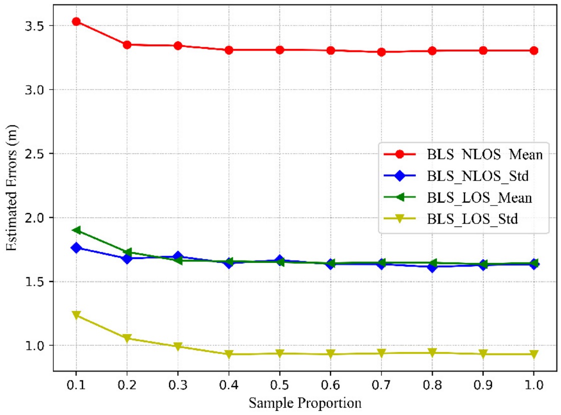


Fig. 10. Relationship between sample number and accuracy.

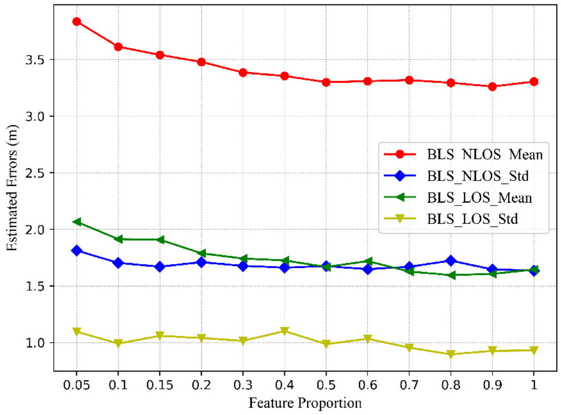


Fig. 11. Relationship between feature number and accuracy.

TABLE V
ACCURACY AFFECTED BY SAMPLING

Algorithms	Laboratory		Meeting Room	
	Mean (m)	Std (m)	Mean (m)	Std (m)
EnsemLoca	**2.5591**	1.4333	**1.18831**	0.8587
EnsemLoca-noSample	2.9415	1.60773	1.35163	0.89125

leads to the decline of the effect. As mentioned above, appropriate sampling improves the accuracy; thus, we need to select the sampling parameters such that they would allow BLS to balance common and separate features.

E. Impact of Sample and Feature Scale

BLS was chosen as the base learner for two reasons. One is that BLS is a neural network, and the other is that its training speed is fast. For a long time, we pay too much attention to the latter one and neglect the former one: its natural attribute. The fact that the BLS belongs to the category of neural networks means that, in the online stage, it is not only advantageous in terms of its processing speed, it also means that BLS is easily affected by different types of samples, which is suitable for the construction of differential predictors. In this experiment, we plan to test the characteristics of BLS in terms of its ability to process samples and features.

BLS was tested in two scenarios, and its performance is shown in Fig. 10. For a single BLS, the more samples, the more accurate are the overall results. Observation of the trend of the curve enables us to identify the smallest proportion of samples, which is between 0.3 and 0.4 of all training samples. After this point, the increased number of samples contributed to all directions of location without any obvious tendency. Hence, it is difficult to further improve the overall situation. The sampling of features is similar to the former; whereas, the convergence point moves significantly to the right side, as shown in Fig. 11. This point is stable after 0.5 and 0.8 for NLOS and LOS scenarios, which also reveals that the characteristics of the two scenarios differ hugely.

We found that feature selection can greatly affect the accuracy of the predictor, whereas sample selection is mainly used to reduce the overall complexity. We choose a position that is slightly worse than the convergence to construct predictors with unique perspectives. This was also proved in our experiment in which we adjusted the parameters; thus, we chose 0.3 as the sampling parameter.

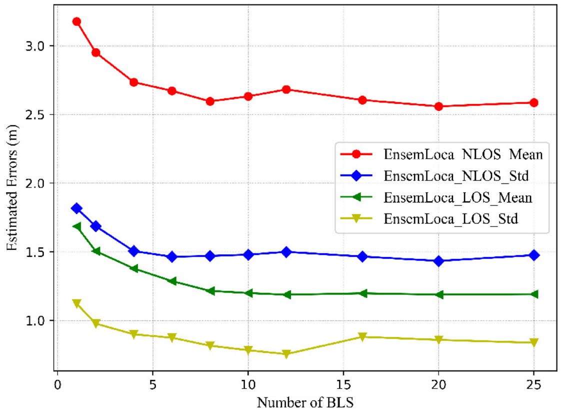


Fig. 12. Relationship between the number of base learner and accuracy.

F. Impact of Number of Base Learners

Finally, the influence of the number of BLSs that constitute EnsemLoca is explored. As shown in Fig. 12, the overall accuracy becomes higher and then reaches a critical value as the number of BLSs increased. This experiment was carried out many times with a low number of BLSs to enable us to describe the trend of change more accurately. Moreover, a sampled BLS was used here because previous experiments proved that this increases the accuracy. In addition, this experiment was also used to guide us to select the number of BLSs as a superparameter.

Fig. 12 perfectly meets our expectation. Because of the use of BLS with strong specificity, EnsemLoca achieves poor results unless the specificity covers all the features. When the types of base learners exceed the threshold, the BLS is responsible for the combination. This includes the combination of their advantages to achieve higher precision. The number of BLSs is a hyperparameter related to the size and the complexity of the scene and the way the BLS is constructed. The larger the scene, the more complex it is, and the more BLSs it needs. Furthermore, the narrower the recognition range of a single BLS, the larger the number of BLSs required, although a more precise result is achieved at the same time.

V. Related Work

Indoor positioning systems have been studied for decades. However, owing to the extensive indoor multipath effect and the unique characteristics of a scene, an effective radio propagation model has not yet been proposed [1], [21]. As the scene changes, the cost and accuracy requirements of the positioning system vary significantly [22]. At the airport or in the shopping mall, rather than using a costly high-accuracy system, rough location estimation is sufficient to meet the positioning needs [23]. Radiofrequency identification (RFID) and ultrawideband (UWB) technology may be a good choice [24] for logistics warehouses, factories, and other scenarios with high-precision positioning needs. Compared with UWB and RFID technologies, which require receivers and transmitters, low-cost Wi-Fi fingerprint technology is sufficient to meet most of the requirements. The ubiquitous Wi-Fi has enabled the related indoor positioning technology to be widely used in public security, business, medical, and other fields.

In a scene with Wi-Fi facilities, depending on the different indoor obstacles and layout, a unique fingerprint database is formed, which is a one-to-one correspondence between the Wi-Fi signal and location. An intuitive idea is to compare the test signal with the signal in the fingerprint database, and then choose the most similar signal' spot as the estimated position. RADAR [4] stores all fingerprints in the offline phase, compares the obtained signals with all stored signals in the fingerprint database in the online matching phase, and then combines the coordinates of the most similar K signals by using the similarity as the weight. Yen *et al.* [25] proposed a coordinated weight-based WKNN to improve the weight relationship of K similar signals. Li *et al.* [26] introduced RSSI-level-based scaling weights into signal vector calculations to further improve the accuracy of KNN. Horus [5] uses a Gaussian regression method to obtain the most possible RSSI. Although, Horus uses a probability method to optimize the fingerprint to enhance the accuracy, the improvement is limited because of signal fluctuation.

In addition, some work chooses ensemble learning to build a positioning system. Fang *et al.* [17] placed eight ZigBee sensors on the ceiling for positioning, which is difficult to deploy and requires the purchase of dedicated sensors. In contrast, Taniuchi and Maekawa [18] made use of the easily accessed RSSI signal and Boosting framework to compensate for the instability of RSSI. However, the predictor trained by Boosting framework has a strict topological relationship, which makes the time consumption of offline and online phases unacceptable. DIFMIC [19] further uses the derivative fingerprints of RSSI to eliminate the disadvantage of the single RSSI signal based on multiple classifiers, while it brings additional complexity to the processing of the source signal.

However, owing to the multipath effect and random noise interference in an indoor environment, the RSSI signal varies widely at the same location. In building a more robust and effective fingerprint database, researchers have focused on CSI signals, which exploit the subcarriers in an OFDM. Using the stable and fine-grained CSI signal collected from the Intel Wi-Fi link 5300 NIC as a fingerprint was shown to improve the accuracy of the positioning system [27], [28]. In addition, researchers who developed a neural network model found that this model not only has faster online positioning performance but is also more accurate. DeepFi [6] used a depth belief network (DBN) model composed of four restricted Boltzmann machines (RBMs) to mine the characteristics of the wireless channel. The complex structure with a large number of neurons makes it impossible for DeepFi to use the general maximum likelihood to train weights. Although the accuracy is very high, it requires more training time than other existing algorithms because the model is composed of RBMs. In contrast, DNNFi [28] is faster than DeepFi because it uses a group of automatic encoders to pre-train the weights layer by layer. Furthermore, the LiFS system [29] found that certain subcarriers in CSI signals are less affected by the multipath effect; therefore, multiple links of CSI signals are used to model the power attenuation equation to improve the positioning accuracy. This also suggests that differences exist in the characteristics of the subcarriers of a CSI signal, and this difference can help us reconstruct the signal characteristics.

Moreover, the development of deep learning models for image recognition has enabled the convolutional neural network (CNN) model to obtain good positioning results. VLocNet [14] proposed global pose regression and odometry estimation from consecutive images based on a CNN model to enhance the accuracy. Xia *et al.* [30] used the Bag-of-Visual-Words model to construct an image database, followed by voting and reclassification. Pedestrian dead reckoning (PDR) technology relies on the inertial measurement unit (IMU) to measure and count the number of steps, step size, and direction of the walker and calculates the walker's walking trace and position information. Generally, PDR technology is combined with Wi-Fi fingerprint technology, with the Wi-Fi fingerprint providing trace compensation for PDR, and PDR revising the Wi-Fi fingerprint location [31]. However, if we do not rely on prior knowledge of the plan, even if the turning error is compensated by road signs, the generated trace error could be as much as 3 m [32]. The SoiCP system [33] uses crowdsensing technology to build a plan to revise the PDR traces. An improvement in the accuracy of the IMU could be expected to increase the accuracy. However, at the same time, the privacy problem has become more prominent and requires us to balance privacy, cost, and performance.

VI. Conclusion

In this study, we built a framework that greatly improves the accuracy without excessively increasing the complexity of the BLS. Furthermore, the time required to test the signal does not increase with the scenario. In the offline training phase, we use a bagging framework to combine BLS as the input of the next layer, where nonlinear combinations are performed. The recognition pattern of the base learner is varied to greatly improve the overall accuracy by using our proposed sampling method for samples and features. In the online phase, the test data are constructed from the same sampling locations, which are sent to the base learners and combinators. The use of layered and sampled architectures further improved the accuracy, even beyond that of a DNN. The performance of the algorithm was proved by data set Widar2.0. Furthermore, we conducted real experiments to demonstrate that the division of labor methodology can be used to guide the framework design.

In future work, we aim to further explore the design of a set of methods to actively assign training tasks to further alienate the base learner and to enhance the accuracy without increasing the computational complexity. In addition, we can further explore the distribution of fingerprint, reconstruct the cluster center of fingerprint, and find the relationship between the number of clusters and predictors. The efficiency of the system is further improved by actively dissimilating and reducing the scale of the predictor.

REFERENCES

[1] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor intelligent fingerprint-based localization: Principles, approaches and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2634–2657, 4th Quart., 2020.

[2] Y. Yu *et al.*, "Precise 3-D indoor localization based on Wi-Fi FTM and built-in sensors," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11753–11765, Dec. 2020.

[3] Z. Yang and K. Järvinen, "The death and rebirth of privacy-preserving WiFi fingerprint localization with paillier encryption," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2018, pp. 1223–1231.

[4] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2000, pp. 775–784.

[5] M. Youssef and A. Agrawala, "The Horus location determination system," *Wireless Netw.*, vol. 14, no. 3, pp. 357–374, 2008.

[6] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[7] Z. Gao, Y. Gao, S. Wang, D. Li, and Y. Xu, "CRISLoc: Reconstructable CSI fingerprinting for indoor smartphone localization," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3422–3437, Mar. 2021.

[8] Q. Song, S. Guo, X. Liu, and Y. Yang, "CSI amplitude fingerprinting-based NB-IoT indoor localization," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1494–1504, Jun. 2018.

[9] N. Chen, T. Qiu, X. Zhou, K. Li, and M. Atiquzzaman, "An intelligent robust networking mechanism for the Internet of Things," *IEEE Commun. Mag.*, vol. 57, no. 11, pp. 91–95, Nov. 2019.

[10] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, p. 53, Jan. 2011.

[11] N. Chen, T. Qiu, C. Mu, M. Han, and P. Zhou, "Deep actor-critic learning-based robustness enhancement of Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6191–6200, Jul. 2020.

[12] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[13] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[14] A. Valada, N. Radwan, and W. Burgard, "Deep auxiliary learning for visual localization and odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018, pp. 6939–6946.

[15] F. Alkhawaja, M. Jaradat, and L. Romdhane, "Techniques of indoor positioning systems (IPS): A survey," in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, 2019, pp. 1–8.

[16] M. Chen, K. Liu, J. Ma, Y. Gu, Z. Dong, and C. Liu, "SWIM: Speed-aware WiFi-based passive indoor localization for mobile ship environment," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 765–779, Feb. 2021.

[17] S.-H. Fang, C.-H. Wang, T.-Y. Huang, C.-H. Yang, and Y.-S. Chen, "An enhanced ZigBee indoor positioning system with an ensemble approach," *IEEE Commun. Lett.*, vol. 16, no. 4, pp. 564–567, Apr. 2012.

[18] D. Taniuchi and T. Maekawa, "Robust Wi-Fi based indoor positioning with ensemble learning," in *Proc. IEEE 10th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2014, pp. 592–597.

[19] X. Guo, N. R. Elikplim, N. Ansari, L. Li, and L. Wang, "Robust WiFi localization by fusing derivative fingerprints of RSS and multiple classifiers," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3177–3186, May 2020.

[20] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, "Widar2.0: Passive human tracking with a single Wi-Fi link," in *Proc. MobiSys*, 2018, pp. 350–361.

[21] F. Liu *et al.*, "Survey on WiFi-based indoor positioning techniques," *IET Commun.*, vol. 14, no. 9, pp. 1372–1383, 2020.

[22] T. Qiu, J. Liu, W. Si, and D. O. Wu, "Robustness optimization scheme with multi-population co-evolution for scale-free wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 3, pp. 1028–1042, Jun. 2019.

[23] V. Djaja-Josko and J. Kolakowski, "A new method for shifted anchor coordinates retrieval in UWB positioning system," in *Proc. IEEE Int. Conf. RFID Technol. Appl. (RFID-TA)*, 2017, pp. 95–99.

[24] V. Gharat, E. Colin, G. Baudoin, and D. Richard, "Indoor performance analysis of LF-RFID based positioning system: Comparison with UHF-RFID and UWB," in *Proc. Int. Conf. Indoor Positioning Indoor Navigation (IPIN)*, 2017, pp. 1–8.

[25] L. Yen, C. Yan, S. Renu, A. Belay, H. Lin, and Y. Ye, "A modified WKNN indoor Wi-Fi localization method with differential coordinates," in *Proc. Int. Conf. Appl. Syst. Innovat. (ICASI)*, 2017, pp. 1822–1824.

[26] D. Li, B. Zhang, and C. Li, "A feature-scaling-based k-nearest neighbor algorithm for indoor positioning systems," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 590–597, Aug. 2016.

[27] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "CSI-based indoor localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1300–1309, Jul. 2013.

[28] G. Wu and P. Tseng, "A deep neural network-based indoor positioning method using channel state information," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2018, pp. 290–294.

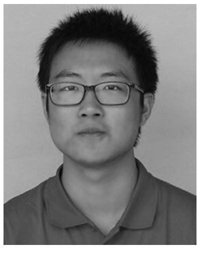
[29] J. Wang *et al.*, "LiFS: Low human-effort, device-free localization with fine-grained subcarrier information," in *Proc. 22nd Annual Int. Conf. Mobile Comput. Netw.*, 2016, pp. 243–256.

[30] Y. Xia, C. Xiu, and D. Yang, "Visual indoor positioning method using image database," in *Proc. Ubiquitous Positioning Indoor Navigation Location-Based Services (UPINLBS)*, 2018, pp. 1–8.

[31] F. Peng and J. Zhai, "A WKNN/PDR/map-matching integrated indoor location method," in *Proc. IEEE 9th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, 2018, pp. 182–185.

[32] Z. Li, X. Zhao, and H. Liang, "Automatic construction of radio maps by crowdsourcing PDR traces for indoor positioning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–6.

[33] Z. Li, X. Zhao, F. Hu, Z. Zhao, J. L. C. Villacrés, and T. Braun, "SoiCP: A seamless outdoor-indoor crowdsensing positioning system," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8626–8644, Oct. 2019.



Chen Wu received the B.E. degree from the Tianjin University (TJU), Tianjin, China, in 2019, where he is currently pursuing the master's degree with the College of Intelligence and Computing.

He is a Merit Student of TJU. He is a member of Tianjin Advanced Networking Key Laboratory. His research focuses on CSI-based indoor fingerprint localization and machine learning.

Tie Qiu (Senior Member, IEEE) received the Ph.D. degree in computer science from Dalian University of Technology, Dalian, China, in 2012.

He held an Assistant Professor and an Associate Professor with the School of Software, Dalian University of Technology in 2008 and 2013, respectively. He is currently a Full Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. He was a Visiting Professor with electrical and computer engineering, Iowa State University, Ames, IA, USA, from 2014 to 2015. He has authored/coauthored nine books, over 100 scientific papers in international journals and conference proceedings, such as IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON MOBILE COMPUTING, TKDE TII, IEEE TRANSACTIONS ON IMAGE PROCESSING, TCY, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *IEEE Communications Surveys and Tutorials*, *IEEE Communications*, INFOCOM, and GLOBECOM. There are ten papers listed as ESI highly cited papers.

Prof. Qiu serves as an Associate Editor for IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, an Area Editor for *Ad Hoc Networks* (Elsevier), an Associate Editor for IEEE ACCESS, *Computers and Electrical Engineering* (Elsevier), and *Human-Centric Computing and Information Sciences* (Springer), and a Guest Editor for *Future Generation Computer Systems*. He serves as the general chair, the program chair, the workshop chair, the publicity chair, the publication chair, or a TPC member of a number of international conferences. He has contributed to the development of three copyrighted software systems and invented 14 patents. He is a Senior Member of China Computer Federation and a Senior Member of ACM.

Chaokun Zhang (Member, IEEE) received the Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2018.

He is currently an Associate Professor with the College of Intelligence and Computing, Tianjin University, Tianjin, China. He was a visiting student with the Department of Computing and Software, McMaster University, Hamilton, ON, Canada, from 2016 to 2017. He has authored/coauthored two books and over 20 scientific papers in international journals and conference proceedings, such as IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE INFOCOM, and IEEE IWQoS.

Dr. Zhang awarded Frontrunner 5000 Top Articles in Outstanding S&T Journals of China in 2017. He is an Internet (CCF TCI) and Network and Data Communications (CCF TCCOMM) professional committee member of China Computer Federation, and a member of ACM, respectively.

Wenyu Qu received the B.S. degree in information science and the M.S. degree in engineering mechanics from Dalian University of Technology, Dalian, China, in 1994 and 1997, respectively, and the Ph.D. degree in applied mathematics from Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2006.

She is currently a Professor with the School of Computer Software, Tianjin University, Tianjin, China. From 2007 to 2015, she was a Professor with Dalian Maritime University, Dalian. From 1997 to 2003, she was an Assistant Professor with Dalian University of Technology. She has authored more than 80 technical papers in international journals and conferences. She is on the committee board for a couple of international conferences. Her research interests include cloud computing, computer networks, and information retrieval.

Dapeng Oliver Wu (Fellow, IEEE) received the B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1990, the M.E. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003.

He is a Professor with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security.

Prof. Wu received the University of Florida Term Professorship Award in 2017, the University of Florida Research Foundation Professorship Award in 2009, the AFOSR Young Investigator Program Award in 2009, the ONR Young Investigator Program Award in 2008, the NSF CAREER Award in 2007, the IEEE Circuits and Systems for Video Technology Transactions Best Paper Award for Year 2001, and the Best Paper Awards in IEEE GLOBECOM 2011 and International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks in 2006. He has served as the Editor-in-Chief for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the Editor-at-Large for IEEE Open Journal of the Communications Society, the founding Editor-in-Chief for *Journal of Advances in Multimedia*, and an Associate Editor for IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, *IEEE Signal Processing Magazine*, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is also a Guest-Editor for IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, Special Issue on Cross-Layer Optimized Wireless Multimedia Communications, and Special Issue on Airborne Communication Networks. He has served as the Technical Program Committee Chair for IEEE INFOCOM 2012, and the TPC Chair for IEEE International Conference on Communications (ICC 2008), Signal Processing for Communications Symposium, and as a member of executive committee and/or technical program committee of over 100 conferences. He was elected as a Distinguished Lecturer by IEEE Vehicular Technology Society in 2016. He has served as the Chair for the Award Committee, and the Chair of Mobile and Wireless Multimedia Interest Group, Technical Committee on Multimedia Communications, and IEEE Communications Society. He was an Elected Member of Multimedia Signal Processing Technical Committee, IEEE Signal Processing Society from January 1, 2009 to December 31, 2012.