**Noname manuscript No.**
(will be inserted by the editor)

# CTSLoc: An Indoor Localization Method Based on CNN by Using Time-series RSSI

**Jingbin Liu 1,2 · Bing Jia 1,2,* · Lei Guo 2 · Baoqi Huang 1,2 · Lei Wang 3 · Thar Baker 4**

Received: date / Accepted: date

**Abstract** Due to the complexities of indoor WiFi signal propagations, it is challenging to improvethe performance of indoor fingerprint-based positioning techniques which is the main hot research in Internet of Things (IoT). Most existing methods have limited positioning accuracy, since they do not take the full advantage of the information available, i.e. timing information attached to the Received Signal Strength Indication (RSSI) vector, and adopt the inappropriate training methods. This paper proposes an indoor localization method based on Convolutional Neural Network (CNN) by using time-series RSSI, termed CTSLoc, by taking into account the correlation among RSSI in time and space. A CNN model is used to extract the temporal fluctuation patterns of RSSI and learn the nonlinear mappings from the signal features with time and space to position coordinates. Finally the trained model is used to predict the user's location. An extensive experiment has been carried out in a space with the size of nearly 1000 squared meters, and a comprehensive comparison with several existing methods indicates that CTSLoc attains a lower average localization error (i.e. 4.23m) and more stable performance than those methods. The CTSLoc method performs relatively less dependent on the amount of data which also eliminates spatial ambiguity and reduces the effect of noise on localization.

**Keywords** Time-series RSSI · WiFi indoor localization · CNN · IoT · IPS

Corresponding author, email: jiabing@imu.edu.cn

1 Inner Mongolia A.R. Key Laboratory of Wireless Networking and Mobile Computing, Hohhot 010021, China

2 College of Computer Science, Inner Mongolia University, Hohhot 010021, China

3 Dalian University of Technology, Dalian 116620, China

4 Department of Computer Science, University of Sharjah, UAE

## 1 Introduction

In recent years, with the development and popularization of Internet of Things (IoT), location based services (LBSs) get a great demand and application prospect. Global positioning system (GPS) has widely developed in outdoor environments while it works in indoor environments [1]. Therefore, great efforts have been devoted to the development of indoor positioning systems (IPS) to achieve reliable and accurate indoor positioning and navigation [1] in the past two decades, meeting people's increasingly demanding needs for indoor positioning. Which is the main and hot research in IoT and IPS area.

At present, many technologies are extensively used in indoor positioning such as [2] RFID, Bluetooth, UWB, WiFi, etc. Among them, WiFi positioning is widely used due to its advantages of convenient deployment, wide coverage, high accuracy, and low cost. Nowadays, WiFi network infrastructures are universally deployed in indoor environments, and almost every ready-made mobile device supports WiFi [1], so RSSI can be easier obtained than Channel State Information (CSI) from most WiFi receivers (mobile devices) [4, 24]. CSI requires specific hardware devices or modified the device driver to acquire it from some advanced WiFi network interface cards [1]. Therefore, indoor positioning based on RSSI is still mainstream.

Due to the complexities of indoor WiFi signal propagations and the device heterogeneity and spatial ambiguity [5], it is challenging to improvethe performance of indoor fingerprint-based positioning techniques. Most existing methods are mainly based on probabilistic methods (i.e. Gaussian distribution, log-normal distribution, etc. [8]), deterministic methods (i.e. KNN, SVM, etc. [10–12]) and neural network methods (i.e. ANN, DNN, etc. [26, 10, 4]). Which mainly use one single RSSI vector for positioning. These methods have limited positioning accuracy, since they do not take the full advantage of the information available, which can able to investigate the relationships inside Received Signal Strength Indication (RSSI) vector, i.e. timing information attached to the RSSI. What's more, some researchers adopt the inappropriate training methods [25].

Therefore, this paper proposes an indoor localization method based on Convolutional Neural Network (CNN) by using time-series RSSI, termed CTSLoc. CTSLoc, taking into account the relationship among RSSI in time and space, is to reduce the limitation of positioning accuracy by attaching timing information on RSSI vector. In CTSLoc, a CNN model is used to extract the temporal fluctuation patterns of RSSI and learn the nonlinear mappings from the signal features with time and space to position coordinates. Finally the trained model is used to predict the user's location. Which attains higher accuracy and better performance than some existing methods.

The main contributions of this paper are as follows.

(1) Better performance on indoor localization: the proposed method CTSLoc takes the full advantage of timing information attached to the RSSI vector in trajectory. Taking into account the correlation among RSSI in time and space which attains better performance than existing methods.

(2) Reducing data noises: the CTSLoc method performs relatively less dependent on the amount of data which also eliminates spatial ambiguity and reduces the effect of noise on localization. The method of data preprocessing is effective in CTSLoc.

(3) Estimation and verification: to estimate and verify the performance and the effectiveness of CTSLoc, there carried out an extensive experiment. Then a series of analyses have given in the following.

The paper is organized as follows: the second section is Related work, which summarizes and introduces the current research methods for WiFi indoor localization. The third section describes the CTSLoc localization model proposed in detail in this paper. The fourth part describes how the model was established and tested, and explains the experimental results in detail. Conclusions will be provided at the end.

## 2 Related Work

In this paper, WiFi-fingerprint-based methods are summarized into the following three categories: probabilistic methods, deterministic methods, and neural network methods.

In the probabilistic approach, the probability density function of RSSI has been assumed a certain distribution of empirical parameters, such as Gaussian distribution, lognormal distribution, etc. [8]. Based on this empirical basis, the literature [9] uses the fitted RSS Gaussian distribution as a location fingerprint to achieve a matching localization between it and the target signal measurements. However, it is not always correct in practice. Compared to other probabilistic systems of the same type, the literature [13] uses a large amount of data for statistical inference to obtain an accurate probability density function, which yields better performance and accuracy. However, it requires a large number of APs for each reference points to obtain a large amount of data, which increases the difficulty and cost of deployment.

In deterministic methods, RSSI is usually used as a feature parameter in combination with a deterministic matching algorithm for location estimation. [10–12] based on KNN method uses a similarity measure to distinguish between fingerprint data and measurement signals in a dataset, and identifies the target point to be measured as the reference point (RP) in the fingerprint library that is closest to its fingerprint in order to determine the node location. The complexity of this algorithm, although low, while does not applicable to unstable indoor environments with the wide fluctuations of RSSI signals. Once the environment changes, the location fingerprint library needs to be rebuilt.

Compared to these algorithms, deep learning methods attain more stable and accurate classification [26]. The literature [10] introduces the extracted information into a shallow neural network to nonlinearly estimate node position coordinates. Literature [15] calibrates the localization results by adjusting the loss function and weights in the CNN model. The authors in [17, 18] propose a traditional DNN-based WiFi fingerprint localization method and experimentally demonstrated that the proposed 4-layer network combined with a hidden Markov model is effective in extracting RSSI signal features and generating initial localization estimates [24].

Table 1 summarizes the main fingerprint-based methods above.

However, the above methods use a single vector of RSSI and do not take full advantage of the information available. The localization accuracy is limited by the noise from individual RSSI readings. Thus, although it has been extensively

**Table 1** Comparisons of main fingerprint-based indoor localization methods

| methods | probabilistic methods | deterministic methods | neural network methods |
|---|---|---|---|
| Characteristic | probability density function of RSSI is assumed a certain distribution of empirical parameters | RSSI is used as a feature parameter in combination with a deterministic matching algorithm for location estimation | nonlinearly estimate node position coordinates with RSSI |
| Advantages | High accuracy | low complexity of algorithm | High accuracy and stable performance |
| Disadvantages | Large scale of data are needed to verify the correction of probability distribution | unstable with the wide fluctuations of RSSI signals, the location fingerprint library needs to be rebuilt Once the environment changes | unappropriate training method is significantly unfavorable for training duration and accuracy |
| Common methods | Gaussian distribution, Lognormal distribution, etc. | SVM, KNN, WKNN, etc. | ANN, DNN, RNN, etc. |

studied, indoor locating based on a single RSSI vector still presents the following problems [19].

(1) Randomly fluctuations in RSSI make the observed location fingerprint data in the testing phase may not match the data in the training phase [6].

(2) The time of data sampling on each reference point is usually short, while most existing methods need a large amount of data.

(3) Some physically distant locations may also have similar fingerprints or fingerprint distances compared to the current location, resulting in ambiguous localization results.

There emerge some localization methods based on trajectory RSSI. The authors in [14] introduce a CNN model that uses time-series RSSI and achieves high accuracy on the classification of multi-building and multi-floor. But the proposed model has no attribute to coordinating estimation [27]. The RNN model proposed in [4] for trajectory localization took into account the correlation between RSSI, which obtained high localization accuracy. In this paper, training trajectories were generated by RPs then a large scale of training dataset is established. However, the RPs were randomly selected by Probabilistic Map which was established by Euclidean distance. Therefore, the correlation between RSSI has uncertainty and may not follow reality.

In order to solve problems in the current indoor positioning methods, this paper proposes an indoor localization method based on CNN by using time-series RSSI, termed CTSLoc, to fill in some gaps of current research.

(1) Taking into account the attaching information on RSSI, timing information has been utilized in the form of time-series to take full advantage of the information available.

(2) Collecting and establishing time-series dataset.

(3) CNN model is used in the area of time-series task of indoor localization.

In CTSLoc, a CNN model is used to extract the temporal fluctuation patterns of RSSI and learn the nonlinear mappings from the signal features with time and space to position coordinates. Finally, the trained model is used to predict the user's location. An extensive experiment has been carried out in a space with the size of nearly 1000 squared meters, and a comprehensive comparison with several existing methods indicates that CTSLoc attainsa lower average localization error (i.e. 4.23m) and more stable performance than those methods [22]. The CTSLoc method performs relatively less dependent on the amount of data, which also eliminates spatial ambiguity and reduces the effect of noise on localization.

## 3 CTSLoc Overview

This paper presents CTSLoc, an indoor localization method based on CNN by using time-series RSSI, which attains a lower average localization error and more stable performance than those methods. Analyzed by the propagation model:

$$RSSI = P_t - K - 10\alpha log_{10}d \tag{1}$$

where $\alpha$ is called the path loss index, $Pt$ is the transmitting power, and $K$ is a constant that depends on the environment and frequency [20]. The attenuation of RSSI fluctuates nonlinearly with time and distance at locations at different distances from the source. User's movement is restricted and the normal speed V is 0.4m/s - 2m/s [21], the distance within the sampling interval $T$ is $S = V * T$. Therefore, in continuous time we are able to obtain continuous trajectories. Considering the correlation among RSSI in time and space in continuous trajectories to preserve the nonlinear fluctuating characteristics. In addition, CNN is highly effective in image problems which has superiority on CTSLoc's dataset. The dataset has two-dimensional matrix constructed from RSSI which is similar to grayscale images.

Based on this, this paper proposes a method for localization using time-series RSSI measurements that use a convolutional neural network model to learn the temporal fluctuation patterns of the time-series RSSI which can be abbreviated as CTSLoc, extract the temporal features of the signal with a nonlinear mapping to time and space, and finally predict the user's location using the established model.

### 3.1 Data Acquisition And Processing

The raw data is a series of one-dimensional vectors consisting of RSSIs collected by the collector from $N$ APs in an experimental environment, defined as $(SSID, MAC, RSSI, T)$, where each $MAC$ represents only an individual specific AP. Since the data collection is based on a predefined path, location coordinates at the start and end points are recorded at each collection. Assume that there are a total of $M$ reference points in the experimental field, $N$ APs, and the acquisition time is $T$. Then the RSSI vector from $N$ APs collected at the $i_{th}$ time point $T_i$ is labeled as:

$$r_N^{t_i} = [RSSI_1^{t_i}, RSSI_2^{t_i}, \cdots, RSSI_j^{t_i}, \cdots, RSSI_N^{t_i}] \tag{2}$$

where $i = 1, \cdots, T$, $j = 1, \cdots, N$ ; $RSSI_j^{t_i}$ denotes the RSSI values from the $j_{th}$ AP collected at the $T = i_{th}$ time point. The corresponding position coordinates are: $loc_i = (loc_{xi}, loc_{yi})$, $i = 1, \cdots, T$, where $loc_{xi}$ and $loc_{yi}$ denote the horizontal and vertical coordinates of the position in space at the $i_{th}$ time point, respectively. Then the total vectors collected in the measured area are expressed as

$$R_N^T = [r_N^{t_1}; r_N^{t_2}; \cdots; r_N^{t_T}] \tag{3}$$

The corresponding matrix of all position coordinates is

$$L_N^T = [l_1; l_2; \cdots; l_T] \tag{4}$$

Conventional methods typically average the collected data over RSSI vectors by location, with each location corresponding to a $1 \times N$ dimensional vector, to construct an $M \times N$ dimensional fingerprint library, where M is a predetermined number of reference points. However, such a method may have large deviations in the averaged data due to the existence of outliers, and a large amount of important information will be lost if multiple RSSI values are averaged, which in addition disrupts the time series arrangement of the data and does not take advantage of the temporal features [23].

Our approach is different, to exploit the temporal characteristics of the signal, the dataset construction method in this paper is to construct a temporal sequence of RSSI vectors. A segment of RSSI trajectory data that is continuous in time is selected and the end position of the segment is used as the coordinates of the entire trajectory. The $T \times N$ dataset is constructed as a time-sequential data set by timestamp, i.e., the $i_{th}$ timestamp is denoted as $t_i$. A sliding window is used to implement the above conception. Assuming that the sliding window size is $win$ and the sliding step is 1, the RSSI feature matrix for the $i_{th}$ time step $W_i$ can be represented as a $win \times N$ matrix as follows

$$\tilde{R}_N^{W_i} = [R_N^{t_i - win + 1}; \cdots; R_N^{t_i - 1}; R_N^{t_i}] \tag{5}$$

The corresponding position coordinates are $loc_i = (loc_{xi}, loc_{yi})$, where $i = win - 1, \cdots, T$. That is, the position corresponding to the RSSI vector at the end of the sliding window. This completes the mapping of the time series RSSI and position coordinates. This method does not take into account the coordinates of the middle position of the trajectory, and the original data are not averaged and randomly selected, which not only ensures the accuracy of the position, but also preserves the RSSI variation and decay characteristics intact, and takes advantage of the time dependence between $T$ and the RSSI readings in the time series. The training set and test set are constructed using the same size sliding window for the RSSI measurements to ensure the consistency of the data scale and to lay the data foundation for the use of CNN positioning.

### 3.2 Data Preprocessing

More than 100 APs were collected in the experimental field, some of which had a large number of missing values or anomalies. Therefore, some abnormal APs need to be cleaned up to avoid the model being contaminated by unstable APs. The RSSI values are standardized to [-100,0], where 0 indicates the strongest signal

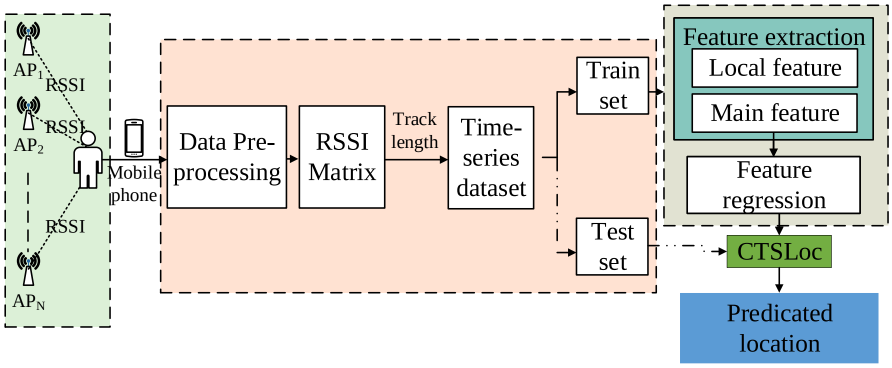



**Fig. 1** The Localization Process of CTSLoc

and -100 indicates a weak signal or no signal. The ones with the smallest missing rate are chosen. By calculating the missing rate the final number of APs selected is 20.

AP chosen is based on the AP missing rate. Suppose that $N_m$ is used to represent the number of elements in the $m_{th}$ column of the matrix $R_N^T$ of $T \times N$ with a value of -100, i.e., the number of samples that represent the missing $m_{th}$ AP. The missing rate $P_m$ of the $m_{th}$ AP is denoted as

$$P_m = N_m/T \tag{6}$$

### 3.3 CTSLoc Model

Specifically, the experimenter continuously collects RSSI data for all RPs in the experimental field at a uniform velocity along a pre-defined trajectory. The collected data are divided by temporal features, marking the start and end points of the trajectory in chronological order, recording the direction of the trajectory, and constructing a data set applicable to the proposed model. And based on the preprocessed data, the CNN model is trained, and the corresponding position label is obtained through feature matching, thus realizing the positioning of the timing RSSI. The localization model conforming to the localization accuracy is obtained and saved. After training, the experimenter continuously collects test data at a uniform speed along the preset trajectory, processes the data in the same way, and calculates the coordinates of the point to be measured. The model structure is shown as Fig. 1. CNN is very prominent in image processing, and the dataset in this paper is processed as a two-dimensional matrix of RSSI of size (win, N) which is equivalent to a grayscale image of the same size. So CNN can better identify the features characteristics in RSSI vector.

In CTSLoc, the CNN finds the non-linear mapping of RSSI to location information by extracting features from the training data. Firstly, the training set is selected to train the CNN model and the whole network is trained using gradient descent and back propagation algorithm, when the loss function between adjacent iterations drops below the threshold or the number of iterations is satisfied,

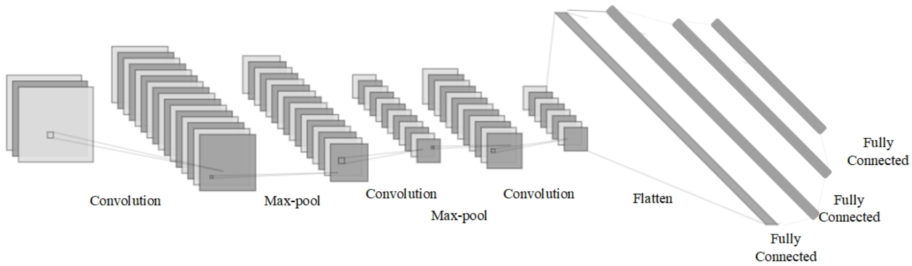



**Fig. 2** The Structure of CTSLoc Model

the network reaches stability then the network parameters are saved, otherwise, a new training sample input from the training sample set is selected to continue the training.

The structure of a CNN generally consists of three parts: the convolutional layer, the pooling layer, and the fully connected layer, and by combining the structure of the three parts in the CNN and adjusting the parameters, the C-NN can be adjusted to be suitable for solving different problems. The CTSLoc model adopts the structure of three convolutional layers, two maximum pooling layers, and three full connection layers. Convolutional operations on the RSSI for feature extraction. The max-pooling layers gradually reduce the space size of the representation. These two parts preserve useful information and reducing the data processing volume which could control overfitting to speed up the convergence of the training network and to obtain new feature expressions. The distortion tolerance energy of the network is improved by feature extraction at the same time. The flatten layer is added to compress the data and connect to the three fully connected layers then obtain the results of the output layer. The model structure is shown as Fig. 2.

The activation function used in CTSLoc after several experiments is rectified linear unit (ReLU):

$$ReLU(X) = max(0, X) = \begin{cases} X, & X > 0 \\ 0, & X \leq 0 \end{cases} \tag{7}$$

It is a segmented linear function commonly used segmented linear function in deep neural networks and performs best in several experiments. It is a unilateral inhibitory function which makes CTSLoc have sparse activation to reduces the interdependence of parameters and alleviates the occurrence of overfitting problems, maintains the convergence of CTSLoc in a stable state to better exploit the relevant features of RSSI. Loss function: the goal of neural networks' training is to minimize the loss function. In the CTSLoc model, the loss function is defined as the Euclidean distance between the output $Y_i$ and the predicted $\tilde{Y}_i$ in the back propagation algorithm:

$$Loss(Y_i, \tilde{Y}_i) = \frac{1}{T}\sum_{i=1}^{T}(Y_i - \tilde{Y}_i)^2 \tag{8}$$

The performance of CTSLoc is calculated by Euclidean distance to measure the performance of localization.

$$D(X_{Pre}, Y_{Pre}, X, Y) = ((X_{Pre} - X)^2 + (Y_{Pre} - Y)^2)^{\frac{1}{2}} \tag{9}$$

Where $X_{Pre}$ and $Y_{Pre}$ denote the predicted coordinates by CTSLoc, $X$ and $Y$ are the measured coordinates in data collecting. $D(X_{Pre}, Y_{Pre}, X, Y)$ is the Euclidean distance between real and predicted coordinates which calculated by (9). The average localization error is defined as:

$$\tilde{D}(i) = \sum_{i=1}^{T} D_i(X_{Pre}, Y_{Pre}, X, Y) \tag{10}$$

In summary, the algorithm of CTSLoc is listed in Algorithm of CTSLoc.

**Algorithm 1** Algorithm of CTSLoc

**Original Data Collecting**: collect original training and testing data based on predefined paths, and record $[SSID, MAC, RSSI, T]$ of each signal;
**Training and testing dataset generating**: denote RSSI vector from $j_{th}$ APs at the $i_{th}$ time point Ti as: $r_N^{t_i} = [RSSI_1^{t_i}, RSSI_2^{t_i}, \cdots, RSSI_j^{t_i}, \cdots, RSSI_N^{t_i}]$;
Establish the datasets:
    $R_N^T = [r_N^{t_1}; r_N^{t_2}; \cdots; r_N^{t_T}]$;
The position coordinates is:
    $L_N^T = [l_1; l_2; \cdots; l_T]$;
**Time Window Data Partitioning**: cut the data with sliding time window of length $win$ and step $\alpha$;
**for** $i$ from $win - 1$ to $T$ **do**
    $\tilde{R}_N^{W_i} = [R_N^{t_i - win + 1}; \cdots; R_N^{t_i - 1}; R_N^{t_i}]$;
    Add $\tilde{R}_N^{W_i}$ to $\tilde{R}_N^W$;
    **return** $\tilde{R}_N^W$;
**end for**
**Data preprocessing**:
**for** $m$ from 1 to $T$ in each columns of $\tilde{R}_N^W$ **do**
    **if** $m == null$ **then**
        $N_m += 1$;
    **end if**
    **return** $N_m$;
**end for**
Missing rate: $P_m = N_m / T$ ;
**if** $P_m > \sigma$ **then**
    Drop $AP_m$ ;
**end if**
**CNN training**: use the preprocessed training data to train the proposed CNN model until loss is stable;
    $Loss(Y_i, \tilde{Y}_i) = \frac{1}{T} \sum_{i=1}^{T} (Y_i - \tilde{Y}_i)^2$;
Save the best model on the hard disk;
**Localization**: Take in testing dataset as an input of the CTSLoc model, and use the output $l = (X_{Pre}, Y_{Pre})$ as a location estimate;

The important symbols and meanings are shown in Tab. 2

**Table 2** Important symbols and meanings

| Symbol | Meaning |
|---|---|
| $RSSI_j^{t_i}$ | RSSI of $j_{th}$ AP at $i_{th}$ time |
| $r_N^{t_i}$ | the RSSI vector of N APs at $i_{th}$ time |
| $R_N^T$ | the matrix of total RSSI vectors |
| $l_i$ | the location coordinates of $i_{th}$ time |
| $L_N^T$ | the matrix of total coordinates |
| $W_i$ | the $i_{th}$ time step |
| $win$ | the length of time step |
| $\tilde{R}_N^{W_i}$ | time-series RSSI matrix of $i_{th}$ time step |
| $P_m$ | the missing rate of $m_{th}$ AP |
| $N_m$ | the available number of RSSI of $m_{th}$ AP |
| $ReLU(X)$ | activation function |
| $Loss(Y_i, \tilde{Y_i})$ | loss function |
| $D(X_{Pre}, Y_{Pre}, X, Y)$ | the Euclidean distance between real and predicted coordinates |

## 4 Experiments

### 4.1 Setup

A large public space withthe size of nearly 1000 squaredmeters on the third floor of Inner Mongolia University Library has been chosen as the experimental space shown in the Fig. 3.

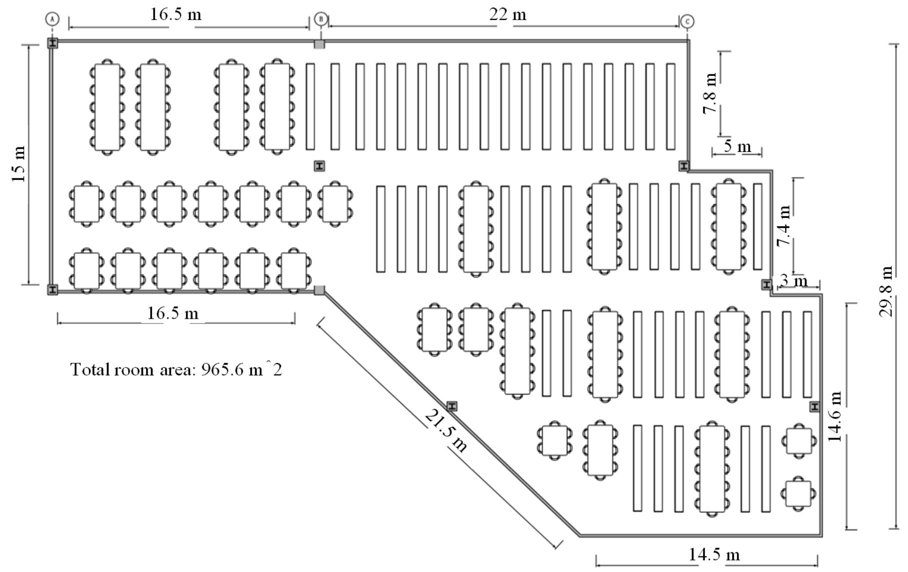



**Fig. 3** Floor Plan of The Experiment Space

There are 45 bookshelves with the height of about 2m, and a number of tables and chairs in the reading area. The target space was divided into regular

**Table 3** The Environment of Software and Hardware

| | | Categories | Details |
|---|---|---|---|
| Software | | Operating System | Windows 10 |
| | | Developing Tools | Pycharm 2020.3.3<br>MATLAB R2017b |
| Hardware | Destktop Computer | CPU | Intel(R) Core(TM) i7-4590 |
| | | RAM | 16GB |
| | | Hard Disk Capacity | 1TB |
| | Laptop Computer | CPU | Intel(R) Core(TM) i5-3337U |
| | | RAM | 8GB |
| | | Hard Disk Capacity | 1TB |
| | Mobile Device | Mobile Phone | Huawei P7 |

grid points spaced at 1m intervals as reference points, and a total of 938 reference points were selected. The train and test set data were collected at different times in scenarios with people walking around irregularly. The experimenter held a smartphone to collect RSSI information by walking along preset trajectories to generate experimental data. Due to the WiFi signal propagations, the speed when data collecting needs to be controlled to approximate a uniform velocity. At the same time, this also makes contributions to accurately estimate according to time and speed.

The experiment has been established on both desktop and laptop, the hardware and software environments are shown in Table 3. And there shows variations on training duration between the two devices.

In addition, the number of APs that can be collected in the experimental environment exceeds 100, and there are some locations where little or no signal can be collected, then the 20 APs with the strongest signals are selected for the experiment. For some signals that cannot be collected at some reference points, the default minimum value of -100 is taken.

### 4.2 Experimental Results and Analysis

To verify the efficiency and superiority of CTSLoc, extensive experiments have been taken place. The experimental results are compared with existing methods and show that CTSLoc obtains lower localization error. The results of ten positioning experiments are shown in Table 4 with an average error of 4.23 m.

**Table 4** Location Errors in Ten Experiments

| NO. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Error(m) | 4.24 | 4.26 | 4.23 | 4.25 | 4.19 | 4.21 | 4.23 | 4.19 | 4.27 | 4.20 | 4.23 |

Both time-series dataset and single-RSSI dataset are used for comparison experiments with CTSLoc which indicates that CTSLoc has lower localization error

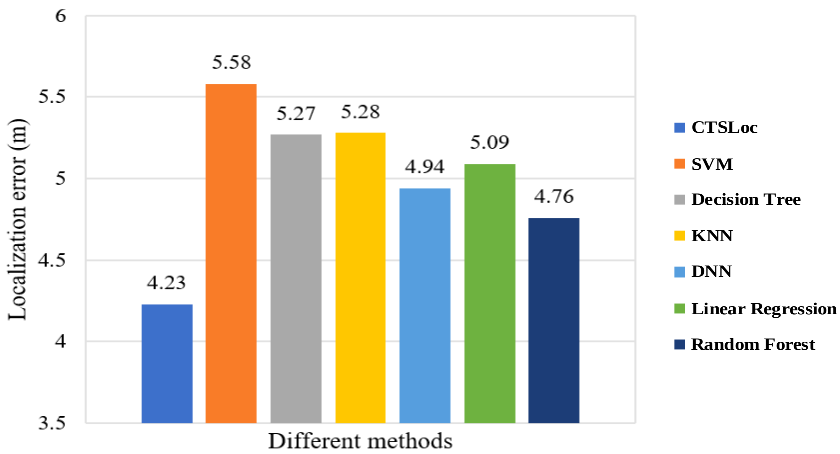
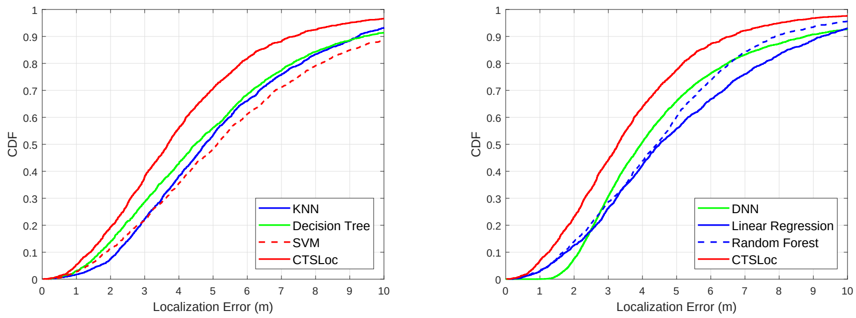



**Fig. 4** The Localization Error of Different Methods



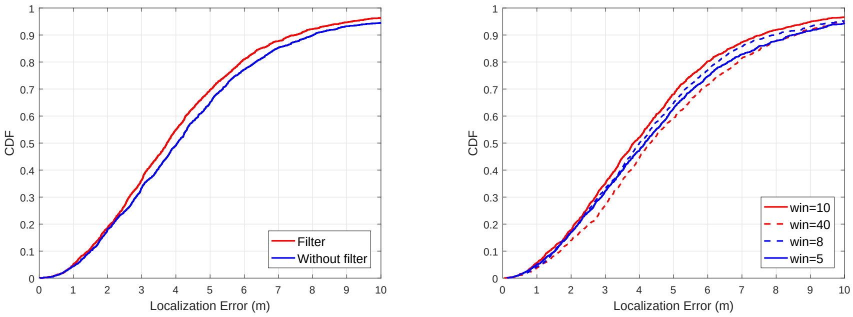

(a) CTSLoc with KNN, Decision tree and SVM

(b) CTSLoc with DNN, Linear regression and Random forest

**Fig. 5** The CDF of the Localization Error of Comparison Experiments

and more stable performance. Results are shown in Fig. 4, which use time-series dataset mentioned in section 3. Fig. 5 shows the CDF of the localization error of CTSLoc with other methods, which use time-series dataset. Fig. 5 (a) shows the localization error of SVM is 5.58m, KNN method is 5.28m and decision tree is 5.27m. In Fig. 5 (b), Random forest reaches a better result of 4.76m while the average localization error of CTSLoc is 4.23m, which is 19.8% and 19.7% and 11.1% lower than them. For a fully comparison, an 8-layer DNN model is established and achieves 4.94m location error. A linear regression method gets 5.09m localization error which is 16.9% higher error than CTSLoc. In comparison, the best results of these methods are used. Therefore, it is significantly showing that, CTSLoc exhibits the lowest positioning error and remarkably improved positioning accuracy.

The efficiency of filter is shown as Fig. 6 (a). The average localization error without filter is 4.52m which is 6% higher than filtered. It is proved that the




(a) (b)

**Fig. 6** The Localization Error of Data-filter and Different Window Sizes

proposed filtering method is effective. Therefore, the experiments in the other part of the paper are completed based on the filtered data.

Fig. 6 (b) shows the comparison of localization results for different time series sliding window sizes, i.e. track lengths of CTSLoc. The best result (4.23m) of sliding window size is 10. The average positioning error increases to 4.81m when the sliding window size is 40. 4.69m and 4.51m when the sliding window size is 5 and 8. Explaining that when the track length increases, the calculated timing information increases, as well as data noises. Therefore, when the track length is longer than the best sliding window size, the effective of data noises is lager than the advantage of time-series information. On the contrary, when the track length is shorter, its timing information is lack for CTSLoc which has no better efficiency than a single RSSI vector.

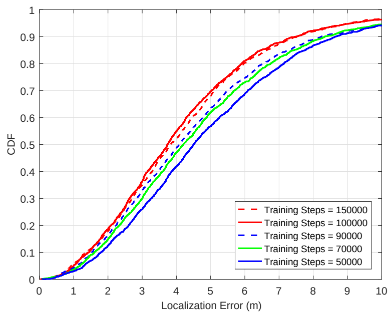



**Fig. 7** The Localization Error of Different Training Steps

Fig. 7 shows the effect of different training steps of CTSLoc. Several experiments have verified that CTSLoc gets the best location result when the training step is 100,000. When it is less than 100,000, the location error decreases with the training steps. The location error is no longer significantly lower and begins to have an upward trend when the training step is more than 100,000, indicating that the CTSLoc network has already fitted the RSSI input pattern and no longer updates the RSSI decay pattern in a gradient, the network is over-fitting. Therefore, the training step that minimizes the location error of CTSLoc is between 90,000 and 100,000 finally determine the training steps to be 100,000. More experimental results are not displayed.

## 5 Conclusions and Future Work

In this paper, a CNN indoor localization model CTSLoc is proposed which considers the timing RSSI and uses the correlation between signals for localization. It solves some of the problems existed in current indoor localization methods such as insufficient data volume, large fluctuations in RSSI and mismatching between training and the testing set, fingerprint localization that is prone to ambiguity, and inaccurate localization results. An extensively experiment has been carried out in a nearly 1000 $m^2$ area for data collecting, and a comprehensive comparison between several existing methods indicates that CTSLoc attains a lower average localization error (i.e. 4.23m) and more stable performance than those methods. The CTSLoc method performs relatively less dependent on the amount of data which also eliminates spatial ambiguity and reduces the effect of noise onlocalization.

There have few limitations on experiment spaces, to adequately demonstrate the superior performance of CTSLoc, more experiments in different spaces would be established in future researches. This paper mentioned the available information attached to RSSI, i.e. timing information. In future works, more kinds of information will be attached to the RSSI vector, there will make greater processes on indoor localization.$\underline{\epsilon_c}$

## Declarations

Fundings

Supported by the National Natural Science Foundation of China (Grants No. 41761086 and 41871363), the Key Science-Technology Project of Inner Mongolia Autonomous Region (Grant No. 2021GG0163) and the Natural Science Foundation of Inner Mongolia Autonomous Region (Grant No. 2021ZD13 and 2019MS06030), the Inner Mongolia Key Laboratory of Wireless Networking and Mobile Computing and the Self-Topic/Open Project of Engineering Research Center of Ecological Big Data, Ministry of Education.

Conflicts of interest

There is no conflict of interest.

Data availability

Data Availability Statements: The dataset used in this manuscript is private data and only available on local machines which can not be shared publicly for now.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Author contributions

All authors contributed equally to the final dissemination of the research investigation as a full article. All authors have read and agreed to the published version of the manuscript.

## References


1. Huang B , Xu Z , Jia B , et al. An Online Radio Map Update Scheme for WiFi Fingerprint-Based Localization[J]. IEEE Internet of Things Journal, 2019:6909-6918.
2. Zafari F , Gkelias A , Leung K K . A Survey of Indoor Localization Systems and Technologies[J]. Communications Surveys & Tutorials, IEEE, 2019.
3. Hoang M T , Yuen B , Ren K , et al. A CNN-LSTM Quantifier for Single Access Point CSI Indoor Localization[J]. 2020.
4. Hoang M T , Yuen B , Dong X , et al. Recurrent Neural Networks For Accurate RSSI Indoor Localization[J]. IEEE Internet of Things Journal, 2019.
5. Chen H , Zhang Y , Li W , et al. ConFi: Convolutional Neural Networks Based Indoor Wi-Fi Localization Using Channel State Information[J]. IEEE Access, 2017, PP:1-1.
6. Liu F , Liu J , Yin Y , et al. A Survey on WiFi-Based Indoor Positioning Technologies[J]. IET Communications, 2020, 14(9).
7. Junlong Z , Chang L , Yang H , et al. Research on WiFi indoor positioning technology based on RSSI[J]. Wireless Internet Technology, 2018.
8. Chen L , Li B , Zhao K , et al. An Improved Algorithm to Generate a Wi-Fi Fingerprint Database for Indoor Positioning[J]. Sensors (Basel, Switzerland), 2013, 13(8).
9. Youssef M , Agrawala A . The Horus WLAN location determination system[J]. Wireless Networks, 2005, 14(3):357-374.
10. S. He and S.-H. G. Chan, Wi-Fi ngerprint-based indoor positioning:recent advances and comparisons, IEEE Communications Surveys & Tutorials, vol. 18, 2016.
11. Mosleh M F , Abd-Alhameed R A , Qasim O A . Indoor Positioning Using Adaptive KNN Algorithm Based Fingerprint Technique[C]// International Conference on Broadband Communications, Networks and Systems. Springer, Cham, 2018.
12. Zhou J , Li W , Jin L , et al. Indoor positioning system based on KNN-SVM algorithm[J]. Journal of Huazhong University of ence & Technology, 2015.
13. C. Figuera, I. Mora, and A. Curieses, Nonparametric model comparison and uncertainty evaluation for signal strength indoor location, IEEE Trans. Mobile Comput., vol. 8, pp. 12501264, 2009.
14. Ibrahim M , Torki M , Elnainay M Y . CNN based Indoor Localization using RSS Time-Series[C]// 2018 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2018.
15. J. Jiao, F. Li, Z. Deng, and W. Ma, A smartphone camera-based indoor positioning algorithm of crowded scenarios with the assistance of deep cnn, Sensors (Basel), vol. 17, Mar. 2017.
16. Wu G S , Tseng P H . A Deep Neural Network-Based Indoor Positioning Method using Channel State Information[C]// 2018 International Conference on Computing, Networking and Communications (ICNC). IEEE Computer Society, 2018.
17. Sinha R S , Lee S M , Rim M , et al. Data Augmentation Schemes for Deep Learning in an Indoor Positioning Application[J]. Electronics, 2019, 8(5):554-.
18. W. Zhang, K. Liu, W. Zhang, et al. Deep Neural Networks for wireless localization in indoor and outdoor environments[J]. Neurocomputing, 2016, 194(C):279-287.
19. Hoang M T , Zhu Y , Yuen B , et al. A Soft Range Limited K-Nearest Neighbours Algorithm for Indoor Localization Enhancement[J]. 2019.

20. Karimi, Hassan A. Advanced Location-Based Technologies and Services[J]. Taylor & Francis, 2013.
21. B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. PickJr, and W. H. Warren, Visual flow influences gait transition speed and preferred walking speed, Experimental Brain Research, vol. 181, pp. 221228, Aug. 2007.
22. Al-Jamimi H A , Al-Roubaiey A . Hybrid Modelling Based on SVM and GA for Intelligent Wi-Fi-based Indoor Localization System[C]// 2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). 2019.
23. Liu G , Qian Z , Wang X . An Indoor WLAN Location Algorithm Based on Fingerprint Database Processing[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2020(13).
24. Z. Xu; Z. Zhang; S. Wang; A. Jolfaei; A. K. Bashir; Y. Yan, and S. Mumtaz. Decentralized Opportunistic Channel Access in CRNs Using Multi-armed Bandits Learning Algorithm. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 5, pp. 57-69, 2021.
25. K. Lou; Y. Yang; E. Wang; Z. Liu; T. Baker, A. K. Bashir. Reinforcement Learning Based Advertising Strategy Using Crowdsensing Vehicular Data. IEEE Transactions on Intelligent Transportation Systems, 2020.
26. F. Qiao; J. Wu; J. Li; A. K. Bashir; S. Mumtaz; and U. Tariq. Trustworthy Edge Storage Orchestration in Intelligent Transportation Systems Using Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems, 2020.
27. K.Z. Ghafoor; L. Kong; S. Zeadally; A. S. Sadiq; G. Epiphniou; M. Hammoudeh; A.K. Bashir; S. Mumtaz. Millimeter-Wave Communication for Internet of Vehicles: Status, Challenges and Perspectives. IEEE Internet of Things Journal, vol. 7, iss. 9, 2020.