Effectiveness of simulation data on walking in Wi-Fi fingerprints using RNN

Shoya Amada¹, Satoru Aikawa¹(а), Shin-ichiro Yamamoto¹

¹Graduate School of Engineering, University of Hyogo
Shosha, himeji-shi, Hyogo 671-2280, Japan
a) aikawa@eng.u-hyogo.ac.jp

Abstract: This study focuses on Wi-Fi fingerprint localization using RNN. Walking survey is a method used for collecting fingerprint datasets. This method can collect continuous walking data while the measurer is walking. However, multiple walking patterns must be measured to accommodate users with different walking speeds and routes. We propose a method for generating simulation data with different walking patterns. The simulation data were created using the coordinate adjacency of the measurement data. Further, we evaluated the model trained with and without the simulation data. Results demonstrate that the accuracy of the proposed method shows its effectiveness.

Keywords: Wi-Fi, Localization, Fingerprint, RNN
Classification: Navigation, guidance, and control systems

References
[1] Ayesha Sarah, Dongsoo Han, “An LSTM-based Indoor Positioning Method Using Wi-Fi Signals” ICVISP, no. 43, pp. 1-5, Aug. 2018.
[2] Rongrong Wang, Haiyong Luo, Qu Wang, Zhaohui Li, Fang Zhao, Jingyu Huang, “A Spatial-Temporal Positioning Algorithm Using Residual Network and LSTM” IEEE Transactions on Instrumentation and Measurement, vol. 69, no. 11, pp. 9251-9261, Nov. 2020.
[3] Sepp Hochreiter, Fakultat fur Informatik, “Long Short-Term Memory” Neural Computation, vol. 9, no. 8, pp. 1735-1780, Jan. 1997.
[4] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink and Jürgen Schmidhuber, “LSTM: A Search Space Odyssey” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, Oct. 2017.
1 Introduction
This study focuses on localization using a fingerprint-based Wi-Fi received signal strength indicator (RSSI). Wi-Fi fingerprints can use existing Wi-Fi access points (APs) and allow localization without knowing the location of those APs. RSSI values are temporally correlated because users move with certain ranges of speeds. Previous studies [1,2] confirmed improvements in localization performance by learning its temporal characteristics with recurrent neural network (RNN). However, users move with various speeds and paths. The patterns in previous studies have been restricted because measuring all the walking patterns is difficult.
We consider various walking patterns of users in localization using an RNN. Here, a tradeoff exists between the number of walking patterns and measurement cost (time). Therefore, we propose generating simulation data with various walking patterns using the adjacency of observation positions (reference coordinates). We aim to reduce the cost and improve the localization performance by training the RNN using the simulation data. The experimental results show the effectiveness of the proposal.

2 Conventional method
The Wi-Fi fingerprinting method is an indoor localization method, which comprises two phases: offline training and online testing. In offline training, a machine learning-based model is trained from the sets of RSSI and its reference coordinate (fingerprinting dataset) is collected in advance. In an online test, the RSSI values are collected in real-time and its location is estimated using the trained model. Point-by-point and walking survey [1] are two methods for collecting fingerprinting data. In the point-by-point, the RSSI values are measured for a specified number of times at each reference coordinate to obtain static fingerprints. In the walking survey, the measurer specifies the route to move and collects time-series RSSI while walking along the route. Position information between two points cannot be obtained because this method measures by determining only the start and endpoints. Thus, we can obtain the information by calculating the walking speed as a constant speed.
Our proposal can create simulation data for the datasets of both methods. However, the walking survey method is similar to the radio environment received when the user is walking. Thus, we used the walking survey method in this experiment.
As mentioned above, the walking survey has the advantage of obtaining time-series data. However, measuring walking patterns comprising all possible walking speeds and paths is challenging because increasing the number of walking patterns requires a longer measurement time. Thus, we address this problem by generating simulation data that simulates user walking.
3 Methodology
This chapter describes the proposal of generating the simulation data and RNN model for learning time-series data.

3.1 Generating Simulation Data
We propose a method for generating simulation data with different walking patterns from data measured with a limited number of walking patterns. The following steps describe the process of creating the simulation data (Fig. 1):

(A) Conversion to Static Fingerprint
Extract static fingerprints obtained at each time from the time-series fingerprints, which are collected using walking survey (Fig. 1 (a)), and obtain the point-by-point datasets. However, the reference coordinates are converted from the start and end time of the measurement as if the measurer was walking at a constant speed.

(B) Generating Simulation Data
(B1) Select P0 (Selected Coordinate) from the list of static fingerprints. Find \(n\) (\(\geq 0\)) neighbor coordinates P1 within a predefined distance from P0 (Fig. 1 (b)). Further, we can obtain \(n\) time-series coordinate data, which consist only of coordinates with the series length of two by combining P0 and P1; perform this process for all coordinates.

(B2) Calculate the vector \(\mathbf{Vec}_{P_0 \rightarrow P_1}\) from P0 to P1 (Fig. 1 (b)).

(B3) Find neighbor coordinates P2 of P1 and the vector \(\mathbf{Vec}_{P_1 \rightarrow P_2}\) from P1 to P2.

(B4) Calculate the difference vector \(\mathbf{V}_d = \mathbf{Vec}_{P_0 \rightarrow P_1} - \mathbf{Vec}_{P_1 \rightarrow P_2}\) and combine
only the coordinates where $|V_d|$ is below a certain value among P2 with time-series coordinate data (Fig. 1 (c)). Set the newly merged coordinate to the selected coordinate.

This process generates data where users do not walk irregularly but move at a constant speed.

(B5) With the number of timesteps $T$, (B2)–(B4) are repeated $(T - 2)$ times to create the time-series coordinate data of lengths $T$. The time-series RSSI data are created by combining the RSSI values observed at each coordinate in the time-series coordinate data that have been created.

As described above, we can expect to improve the performance for unknown walking patterns by creating the simulation data as described above. However, performance may be reduced by different movement directions of the data to be combined because there is an effect of shielding by the human body. We will experimentally confirm effectiveness of the simulation data, including this negative influence.

In the following, the data before simulation data is generated is called the original data.

### 3.2 RNN-Based Method

#### 3.2.1 Applying RNN

To study time-series data, we use RNN and long short-term memory (LSTM) as estimation methods. LSTM is introduced to solve the gradient vanishing problem of RNN [3].

We preprocess data, making it suitable for training the model. First, considering that the range of the RSSI values is $-100$ to $0$ dBm, we set the missing Wi-Fi signals to $-100$ dBm. Further, we use min–max normalization to normalize RSSI values from zero to one.

#### 3.2.2 RNN Structure

The input data of RNN is $\text{RSSI}^{(t_x,t_{x+T})}$ from $t = t_x$ to $t = t_{x+T}$, and the output is the value of the coordinate $e^{t_{x+T}}$ at $t = t_{x+T}$. Here, the number of timesteps $T$ determines series length from $t_x$ to $t_{x+T}$, and the greater the $T$ value, the more information from the past can be incorporated. Further, we employed a stacked RNN structure, which comprises multiple RNN layers. Stacked RNN structures allow deeper structures to be created using the RNN’s hidden state as input to another RNN [4]. The loss function uses MSE, which comprises the error distance between the estimated coordinate $e^{t_{x+T}}$ and correct coordinate $e^{t_{x+T}}$.

### 4 Experiments

In this chapter, we conducted several experiments to confirm the effectiveness of the simulation data.

#### 4.1 Data Collection

We measured time-series fingerprints in a building hallway while walking...
holding a device in hand (Fig. 2). Further, we measured the training data by making four round-trips of three straight lines at a constant speed (=approx. 1 m/s) on both sides (Fig. 2 (a)). For the test data, we measured not only the walking pattern of the training data but also different walking patterns, such as clockwise or zigzagging (Fig. 2 (b)), and increased the speed to approx. 1.5 m/s. The difference between these walking patterns is that the effectiveness of the simulation data is experimentally confirmed. The coordinates of the measured training and test data were 837 and 130, respectively. The number of APs observed in all coordinates was 147, which were used for model training. Further, we generated the simulation data with an arbitrary length \( T \) from the measured data. The distance that defines neighbor coordinates described in Section 3.1 was set to approximately 3 m. This distance was calculated as 1.5 m/s \( \times 2 \text{ s} = 3 \text{ m} \), considering that the maximum speed of users is approximately 1.5 m/s and the reception interval of the device is approximately 2 s. The number of samples of the original and simulation data obtained from 837 points was 789 and 76651 samples, respectively, at \( T = 3 \). Note that the original data were obtained by slicing the measured time-series fingerprints by specified timesteps.

| (a) Training data | (b) Test data |
|-------------------|--------------|

Fig. 2. Measurement environment

### 4.2 Performance of Our Proposed Algorithm
We built each RNN (LSTM) model-trained simulation and original data. Further, we varied the number of RNN layers and timesteps \( T \) from one to two and three to five, respectively, and examined the estimation performance on the test data. We compared the highest performance at each \( T \) using average and maximum errors as evaluation indicators.

The model’s performance using the simulation data improves at each \( T \) for the average and maximum errors (Fig. 3 (a)). Further, the model’s performance using the simulation data improves as \( T \) value increases, whereas the performance of the model using the original data decreases. This may be because of the differences in walking patterns, particularly the walking speed, between the training and test data.

Next, we experimented with different conditions to determine the effects of different walking speeds. The test data were classified into two types of walking speeds, which are normal (approximately 1 m/s) and fast (approximately 1.5 m/s), and we evaluated four models. The four models are two RNN and two LSTM models, which were evaluated using the original and simulation data. Each model is the one at the highest performance obtained in the above experiment. Figs. 3 (b) and (c) show the CDF of localization errors for normal and fast speeds,
respectively. Fig. 3 (b) shows that the RNN model trained with the original data has the highest performance for the test data with normal. This is because there is no difference in speed between the training and test data. However, Fig. 3 (c) shows that the two models using the simulation data provide higher performance for the fast speed. These results show that the model using the simulation data provided high performance even for data with walking patterns that were not measured.

![Graph showing performance comparison](image)

(b) CDF of error with the normal speed  
(c) CDF of error with the fast speed

**Fig. 3.** Comparison of the original and simulation model

## 5 Conclusion

This study proposed methods for generating simulation data that simulate the changes in Wi-Fi signals caused by various users’ walking for fingerprint localization using RNN. Simulation data were created with different walking patterns by processing the measured data using the adjacency of the reference coordinates, thereby reducing the measurement cost. By training the RNN model using the simulation data, we obtained a high performance for the test data with various walking patterns.

**Acknowledgments**

This work was supported by JSPS KAKENHI Grant Number 21K04065.