An Improved Wireless Positioning Algorithm Based on the LSTM Network

Xiansheng Yang, Dong Chen, Jianzhu Huai, Xiaoxiang Cao, and Yuan Zhuang(✉)

The State Key Laboratory of Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China

yuan.zhuang@whu.edu.cn

Abstract. Given that the BDS-3 (Beidou System-3) has been accomplished and works well, there are increasing demands for localization and navigation in daily life. However, BDS-3's signals cannot cover some challenging areas such as urban canyons and indoor environments. To extend the availability of the navigation system, other positioning technologies are required to aid the BDS-3. Wireless fingerprint localization technologies (e.g., Wi-Fi, Bluetooth, 5G, etc.) have attracted lots of attention worldwide due to their ubiquitous and cost-effective characteristics, but there are various challenges such as fingerprints spatial ambiguities, RSS (Received Signal Strength) fluctuations over time and RSS variation caused by devices heterogeneity, which impairs positioning accuracy and precision. By analyzing the relationships hidden in adjacent fingerprints, we utilize the Encoder-Decoder Framework and the sequence-based Long Short-Term Memory (LSTM) network to convert vulnerable RSS to stable RSS spatial gradient, which can eliminate RSS fluctuation over time and hardware diversity. The sequence-based LSTM also eliminates fingerprint spatial ambiguities using the sequence match. The preliminary experiments show the superiority of the proposed framework over the-state-of-art methods in terms of robustness and precision. Specifically, the proposed framework reduces average positioning errors by 24.68% and decreases the average errors by 36.38% and 6.8% in terms of the resistance to device diversity and RSS fluctuation over time respectively.

Keywords: Deep learning · LSTM · Fingerprint · Wireless positioning

1 Introduction

With the advancement of urbanization, there are increasing numbers of large buildings such as urban complexes and shopping malls, which not only block the signal of the Global Navigation Satellite System (GNSS) but also expands people's demands for indoor positioning.

There are various existing signals available for indoor positioning, including Wi-Fi [1], cellular networks [2], Radio Frequency Identification devices (RFID), Ultra-Width Ban (UWB), visible light [12] and so on. Among all above technologies, Wi-Fi-based wireless technologies are relatively popular due to their ubiquitous and cost-effective characteristics, but they are also faced with many challenges, such as RSS fluctuation

C. Yang and J. Xie (Eds.): *China Satellite Navigation Conference (CSNC 2021) Proceedings*, LNEE 773, pp. 616–627, 2021.
https://doi.org/10.1007/978-981-16-3142-9_59

over time and RSS variation caused by device diversity. The former is that the RSS will be very different on the same point after a long-time interval, which cost people a lot of money and resources to frequently update fingerprint in the database to keep high positioning precision, and the latter is that RSS collected simultaneously by two different brands of devices are also very different, which is another serious problem.

Most methods [13, 14] only achieve high precision in some specified scenarios, but fingerprint spatial ambiguity, fingerprint fluctuation over time, and fingerprint variety caused by device diversity still exist, which impaired their precision seriously. So most of them may not work well on real problems.

In this work, we have achieved a robust positioning algorithm utilizing the Encoder-Decoder framework to fuse fingerprints spatial gradient [10] and fingerprints. Hidden sequential features and hidden sequential gradient features are extracted from adjacent fingerprints in the Decoder Module and adjacent fingerprint spatial gradients in the Encoder Module respectively, and then are matched with the fingerprints in the database by LSTM. This will alleviate fingerprints instability caused by devices heterogeneity and fingerprints fluctuation over time. The main contributions of this paper are as follows:

- The proposed algorithm fuses fingerprints and fingerprint spatial gradient using the Encoder-Decoder framework, which can effectively alleviate fingerprint fluctuation over time and fingerprint instability caused by device diversity.
- We extract the sequence information hidden from adjacent fingerprints and adjacent fingerprint spatial gradient respectively, and then match fingerprints in the database, which can eliminate the spatial ambiguity of fingerprint and improve positioning precision.

The rest of this paper is organized as follows. After reviewing related work in Sect. 2, we present an overview of the proposed model and extensive details of every part of the algorithm in Sect. 3. results based on experimental trials are discussed in Sect. 4. Section 5 concludes the paper.

2 Related Work

Traditional fingerprint technologies, such as Nearest Neighbour and K-Nearest Neighbour algorithm [15], calculates the similarity between fingerprints from the database and current position, and then assigns different weights to K nearest positions, so the final position is a weighted average of the K nearest positions. Its time complexity is O(2), which is slow to navigate on a large database in a real-time manner. Mirowski et al. [4] calculated the similarity between fingerprints with time-effectively Kullback–Leibler divergence, but they were also faced with many problems, such as fingerprint fluctuation and spatial ambiguity. Recently, with the increase of the capacity of the GPU (Graphics Processing Unit), some methods from the Artificial Intelligence have become popular. You et al. [5] applied DRL (Deep Reinforcement Learning) to indoor positioning, which provided a new viewpoint. Qun et al. proposed a new model, named Deep Navi [6], which projected various information including geomagnetism, Wi-Fi, and visual image into a common space and then put these features into MDN (Mixed Dense Network) to

infer current position. Instead of using traditional single-point matching, Hoang et al. [7] used trajectory data for matching, which eliminated questions caused by the RSSI short collecting time per location during positioning and the authors also compared the performances of different Recurrent Neural Networks (RNN). All of the above methods could only achieve high precision in some specified scenarios, but fingerprint spatial ambiguity, fingerprint fluctuation over time, and fingerprint variety caused by device diversity still exist.

3 Positioning Algorithm Using Encoder-Decoder Framework

3.1 Data Processing

The data processing program consists of two parts, the offline stage and the online stage, as shown in Fig. 1. In the offline stage, as shown in black arrow in Fig. 1, we transform RSS collected by devices into fingerprints stored in database and optimize the model's parameters. The entire path is divided into s RP (Reference Points) and the distance between adjacent reference points is d. In this work, the data are collected continuously; that is, a person with the device passes the trajectory at a constant speed and then receive $RSS_i = \left\{rss_i^1, \ldots, rss_i^j, \ldots, rss_i^m\right\}$ rss_i^j means that the RSS is received from j^{th} AP at time t_i and position $pos_i = \{x_i, y_i\}$ at time $t_i\{\mathrm{i} = 1, 2, \ldots\}$. Then, we can obtain a fingerprint database $\mathrm{F} = \{f_1, f_2, \ldots, f_n\}$, where $f_i = \{t_i, pos_i, RSS_i\}$ shown in Fig. 1. After the fingerprint database is generated, we train our model and optimize the parameters. Firstly, the data passes through the Window Split Module, forming a fingerprint sequence. Then the fingerprint sequence is put into the Decoder Module and the Gradient Module, respectively.

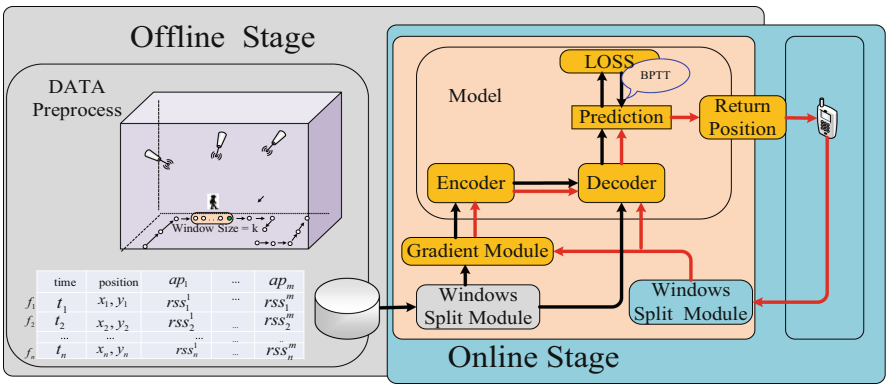

Fig. 1. The overall workflow of the proposed method

The latter processes fingerprint sequence into fingerprint spatial gradients. The fingerprint spatial gradients are put into the Encoder Module where the sequential information is extracted and form the hidden state. Then the output of the Windows Split

Module and hidden states of the Encoder are put into the Decoder Module together, so the Prediction Module will output the current position. We calculate errors between the real location label and the output of the Prediction Module with the cross-entropy loss function, which will be used to update the parameters of the network through the Back-propagation Through Time (BPTT) algorithm. In the online stage as shown in the red arrow in Fig. 1, the data processing program is similar to the offline stage where the differences are that the output of the Prediction Module is send to users on the online stage directly rather than updating parameters of model on the offline stage.

3.2 Extract Features

In the data processing program, the fingerprint database has been established as shown in the table in Fig. 1. The extracting features program is made up of the Window Split Module, the Gradient Module and the Public Module.

3.2.1 The Window Split Module

This module mainly allocates the data into different windows. Assuming that the windows size is k in a trajectory, we have gotten fingerprint f_p at t_p, as shown in Fig. 2, and then extract k–1 fingerprints from the previous fingerprints in a time order, all of which will form $w_p = \{f_{p-k+1}, \ldots, f_p\}$. We can get $\mathbf{W} = \{w_k, w_{k-1}, \ldots, w_n\}$ in a trajectory and note the subscript of w begins with k because the size of window is k.

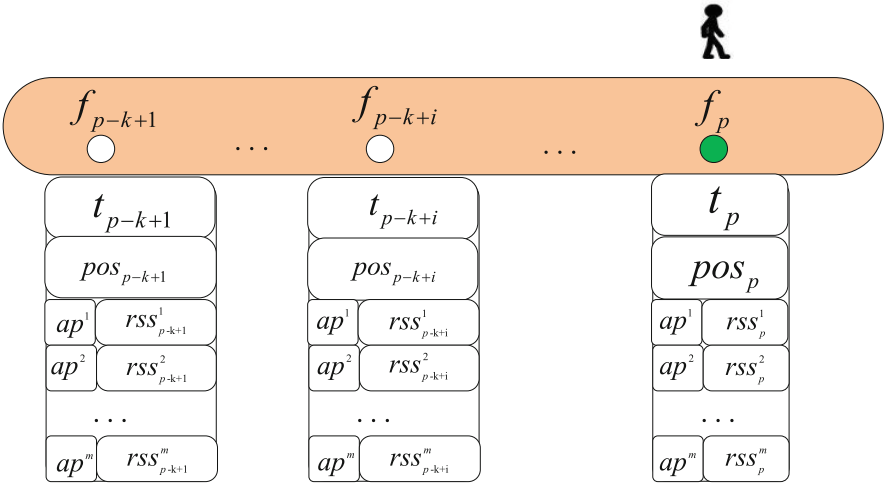

Fig. 2. Windows split module

3.2.2 The Gradient Module

Although the RSS at the same location changes over time, the RSS differences between adjacent locations will be relatively stable and will not change rapidly over time [8].

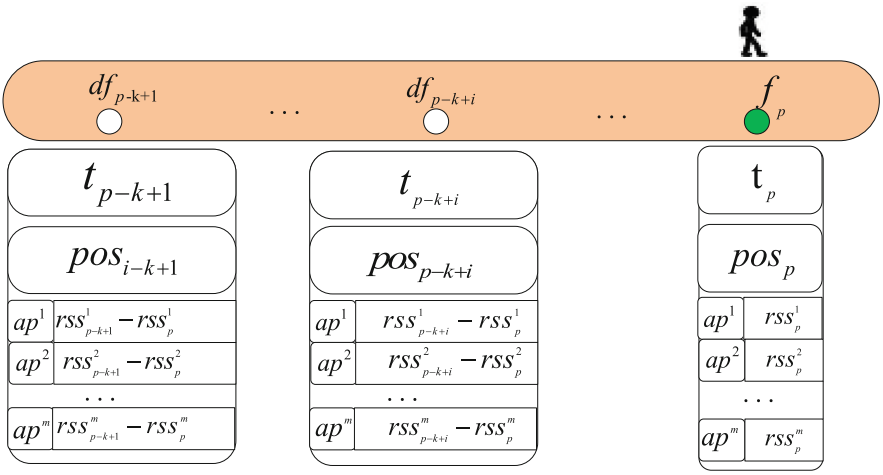

Fig. 3. Gradient module

In addition, subtraction between RSS of adjacent location can eliminate the bad effect of devices diversity [3]. If the same trajectory is passed twice at a long-time interval and $T^1_{(i,j)}$ is the RSS from j^{th} AP at i^{th} RP for the first time. After a period of time, we collect $T^2_{(i,j)}$ at the same position for the second time. $(T^1_{(i,j)} - T^2_{(i,j)})$ is very large, indicating that RSS changes rapidly over time and fingerprints match with RSS is vulnerable. However, the difference between $(T^1_{(i,j)} - T^1_{(i-1,j)})$ and $(T^2_{(i,j)} - T^2_{(i-1,j)})$ is little, implying fingerprint spatial gradient is stable. Moreover, two devices collect RSS simultaneously and $S^1_{(i,j)}$ is the RSS that first device received from j^{th} AP at i^{th} RP as well $S^2_{(i,j)}$ is the RSS that second device received from j^{th} AP at i^{th} RP. The $(S^1_{(i,j)} - S^2_{(i,j)})$ is totally different, which would dramatically impair the precision of positioning. However, the difference between $(S^1_{(i,j)} - S^1_{(i-1,j)})$ and $(S^2_{(i,j)} - S^2_{(i-1,j)})$ is little, which means the fingerprint gradient will eliminate diversity of different brands of devices. From the above analysis, we define fingerprint spatial gradient dw_p at t_p, which can be calculated from w_p directly as shown Fig. 3. From w_p to dw_p, t_{p-k+i} (i = 1, 2…, k) and pos_{p-k+i} (i = 1, 2…, k) are invariant but RSS_{p-k+i} will be transformed to $DRSS_{p-k+i}$. We define $DRSS_{p-k+i} = \left\{drss^1_{p-k+i}, drss^2_{p-k+i}, \ldots, drss^m_{p-k+i}\right\}$ where $drss^j_{p-k+i} = \left(rss^j_{p-k+i} - rss^j_p\right)$ and i = 1,2…, k−1; In other words, the subtraction between the RSS at current position and the RSS at end of Window from the same AP. So we can get $df_{p-k+i} = \left\{t_{p-k+i}, pos_{p-k+i}, DRSS_{p-k+i}\right\}$, but note that $dw_p = \left\{df_{p-k+1}, df_{p-k+2}, \ldots, f_p\right\}$, where the $df_p = f_p$ at a window. Finally, we can get all fingerprint spatial gradient $\mathbf{DW} = \{dw_k, dw_{k+1}, \ldots, dw_n\}$.

3.2.3 The Public Module

Three types of the Public Modules: **A**, **B**, and **C** are presented in Fig. 4. They consist of different MLP (Multi-Layer Proceptions) and these Modules in the same type share

their parameters with each other, which means there are only three set of parameter values. Since RSS, DRSS and $\hat{y}$ have different measurement scales, it is unreasonable to concatenate or send them to other Modules (such as the Encoder Module or the Decoder Module) together so we use the Public Module to solve this problem.

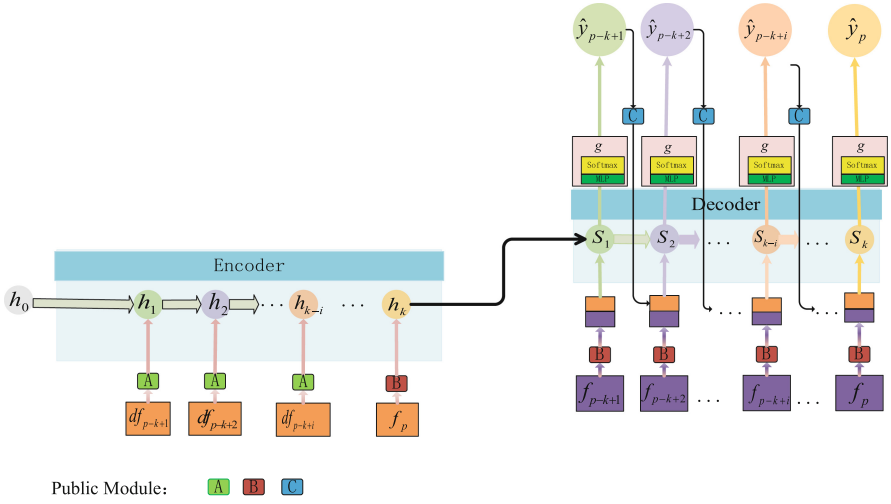

Fig. 4. Framework of the proposed algorithm

3.3 Algorithm Framework

The Encoder-Decoder Framework [9] is one of the most prevalent frameworks in the deep learning field and perform well in solving many problems. It consists of two parts: The Encoder Module and the Decoder Module. In the Encoder Module, we should design an appropriate neural network to extract features from the input data, acquiring hidden semantic; While in the Decoder Module, we also design a neural network to absorb hidden semantic produced by the Encoder Module and other factors, which is used to predict current position. LSTM [10] is a kind of RNN and is equipped with excellent "memory" because of its three logic gates (the forgetting gate, input gate and out gate). The good "memory" allows it to remember previous information from a long time ago and avoid gradient disappearance problem [11] that stop parameters of neural network updating. This paper fuses fingerprint spatial gradients and fingerprints by Encoder-Decoder framework as shown in Fig. 4, which can alleviate the effect of device diversity and RSS fluctuation over time effectively in the positioning system. Considering hidden sequential information in adjacent fingerprints and adjacent fingerprint spatial gradients, we hire LSTM Cell in the Decoder Module and the Encoder Module to abstract hidden sequential information, which relieve the space ambiguity of fingerprint effectively and improve localization precision to some extent. Specifically, fingerprints database F pass through the Window Split Module and the Gradient Module, generating sequential fingerprints database **W** and sequential fingerprint spatial gradients **DW** (Sect. 3.2)

respectively. We take out w_i from **W** as well dw_i from **DW** and then put them into the Decoder Module comprised of multiple LSTM Cells as well the Encoder Module comprised of multiple LSTM Cells respectively. Each LSTM Cell will output cell state and hidden state**,** both of which have the same dimension and are passed to the next LSTM Cell. For convenience, all cell state and hidden state in the Encoder Module and in the Decoder Module are represented by (c_i^e, h_i), where i = 0, 1, 2.., k and (c_i^d, s_i),where i = 0,1,…, k, respectively, and we also set h_0, c_0^e = the matrix consisted of zero and $(c_0^d, s_0) = (c_k^e, h_k)$. So, we can define the Encoder Module as follows:

$$\left(\mathrm{H},\ C^e\right) \ = \ \text{Encoder}\ (DRSS_{p-k+1},\ \ldots,\ DRSS_{p-k+i},\ \ldots,\ DRSS_p) \tag{1}$$

where $\mathrm{H} = \{h_1, \ldots, h_k\}$ is the hidden state containing sequential information extracted by LSTM Cell from DRSS and $C^e = \{c_1^e, \ldots, c_k^e\}$ is cell states; the Encoder is a neural network constituted of k LSTM Cells that pass massages to each other by hidden states h_i and cell state c_i^e; $DRSS_i$ is fingerprint spatial gradients from dw_i.While we define the Decoder Module as follows:

$$\left(c_i^d,\ s_i\right) \ = \ \text{Decoder}\ (\hat{y}_{i-1},\ s_{i-1},\ RSS_i) \tag{2}$$

where s_i is hidden state containing sequential information extracted by LSTM Cell from RSS; $\hat{y}_{i-1}$ is output of the Prediction Module at previous time, namely previous position, which also affect the prediction of position at current time. s_{i-1} is the previous hidden state in the Decoder Module and RSS_i is fingerprint at current time. Then, we can predict current position as follows:

$$\hat{y}_i = \mathbf{g}(\hat{y}_{i-1},\ s_i,\ RSS_i) \tag{3}$$

$\hat{y}_i$ is the output and the **g** is the Prediction Module consisted of full-connected layers as well the SoftMax layer. Here we need to discuss the input data of the Encoder Module and the Decoder Module. The input of proposed model consists of two parts, RSS and DRSS. The reason why we input RSS into the Decoder Module is that the relationship between fingerprint and position is more direct, and we can extract the hidden features by LSTM Cells in the Decoder Module. However, the relationship between the fingerprint spatial gradients and the position is more difficult to discover but have pivotal effect on the positioning system especially in the case of complex scenes. We conduct a series of experiments to prove this idea.

4 Experiment Evaluation

4.1 Data Description

In order to evaluate the performance of the proposed algorithm, we conduced various experiments on a test site with an area of 4859 m^2 (113 m * 43 m). The map is shown in Fig. 5. At the same time, in order to validate the capacity of the model relieving device diversity, we use a total of 4 devices to collect data, including Samsung S5, Xiaomi Mi4_1, Xiaomi Mi4_2 and Xiaomi Mi4 black, and they are divided into two

groups: $group_1$ consists of Samsung Galaxy S5 and Xiaomi Mi4_2 where 20 trajectories are collected; $group_2$ are made up of Mi4_1 and Mi4_black where 24 trajectories are collected. Two volunteers use a continuous collection method to collect data, which means volunteers go through all trajectories at a constant speed, and the RSS, current time t_i as well location coordinates pos_i are recorded when they arrive at each RP.

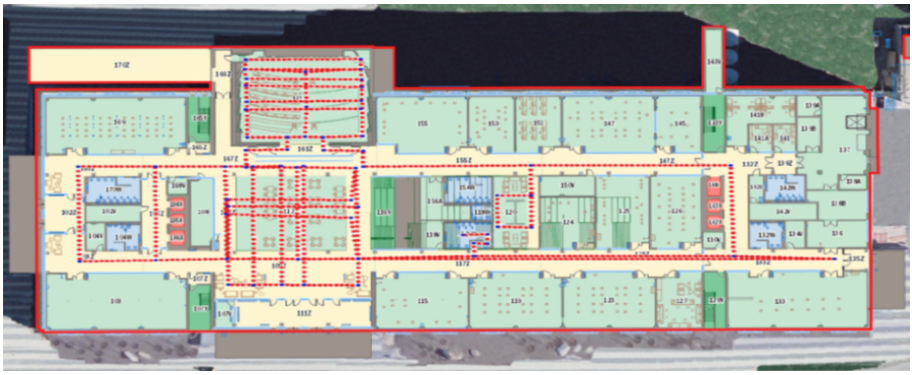

Fig. 5. The trial site in our experiment

4.2 Software and Hardware Equipment

All the baseline and the proposed model are implemented on a server. We use two NVIDIA GeForce RTX2080Ti image processing units with 10 GB memory. For hyper-parameters, we set the learning rate LR $= 10e^{-4}$ and use the SGD optimizer. The batch size is 100 and the model training phase cost 1,681.81 s; while the prediction phase cost 0.01s for a single sample. We set dropout $= 0.5$ to avoid overfitting.

4.3 Model Comparison

In order to verify the performance of the proposed model, we compared the proposed method with K-Nearest Neighbors algorithm, Support Vector Regression (SVR), Random Forest (RF) and xgboost. Cumulative Distribution Function (CDF), Root Mean Square (RMS) and running time are used as metrics. If not specified, the time sequence length of the proposed method is 4 (window size or TIME STEP $= 4$) and the grid size is 3 m.

4.3.1 Comparison Schemes on Same Devices

We have conducted extensive experiments on four data sets. In this part, test data and training data are collected by the same device, and then the RMS are calculated. The RMS of the proposed framework decrease about 1 m (3.57 m), compared the traditional method (4.74 m). Although the training phase of this method takes the majority of the time (1681.81 s), It only costs about 0.01 s to predict a single position in the online stage,

which is acceptable in most scenarios. We also separately compared the performance of different methods on the four data sets. The CDF curves of different methods on the four data sets can be found in Fig. 6 and the proposed method is better than other methods, especially in test data of S5 (4.06 m) and Mi4_2 (4.31 m), whose environment is more complex.

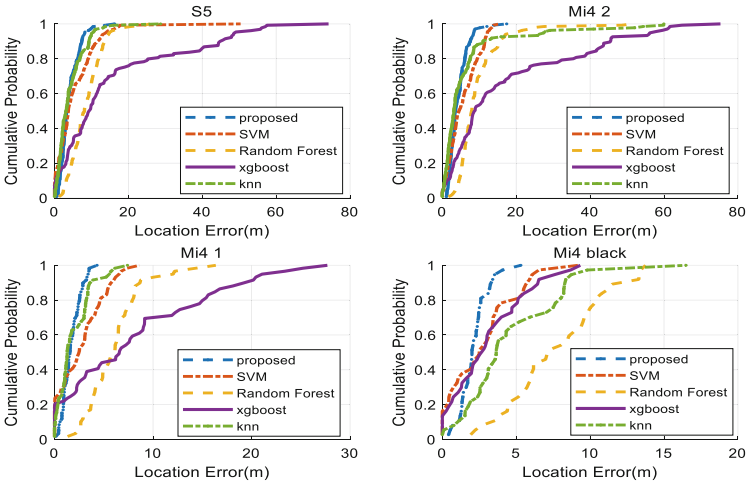

Fig. 6. CDF of location error on four datasets

4.3.2 Comparison Scheme on Different Devices

In order to test the anti-hardware interference ability of the proposed model, our training data and test data are collected by different devices. The four mentioned devices are divided into two groups. In the first experiment, training data and test data were collected by Mi4_1 and Mi4 black respectively. It can be seen that the RMS of the model in this work (3.73 m) is the lowest, and RMS increased 1.58 m (it was the smallest among all methods) compared with experiments on the data collected by the same devices, indicating the strongest robustness for hardware interference. And Fig. 7 is the cumulative distribution function (CDF) of this experiment. In the second experiment, as shown in Fig. 8, the training data was collected by Mi4_2 and the test data was collected by S5. It can also be found that the proposed algorithm in this framework has the smallest RMS (5.60 m) and the rise in RMS is also the smallest (1.54 m).

4.3.3 Hyper-Parameters Analysis

This section extends detailed extensive experiments of choosing optimal the length of the TIME STEP on the data set collected by Mi4_2. As shown in Fig. 9, the left y-axis is training time while the right y-axis is RMS, and the abscissa is the TIME STEP. Because the time of prediction in the actual scene does not change much with the TIME STEP, it is not necessary to describe the test time in detail. It can be seen from Fig. 9 that when the

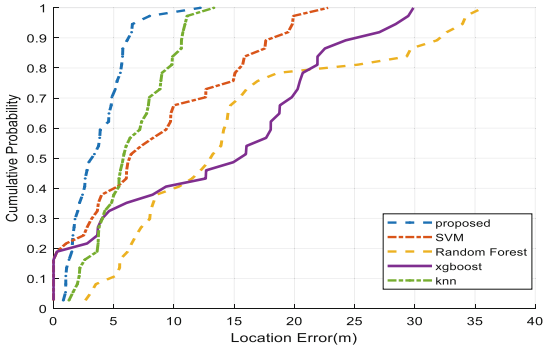

Fig. 7. CDF of location error to evaluate robustness in terms of hard ware. Training data and test data collected by Mi4_1 and Mi4 black, respectively

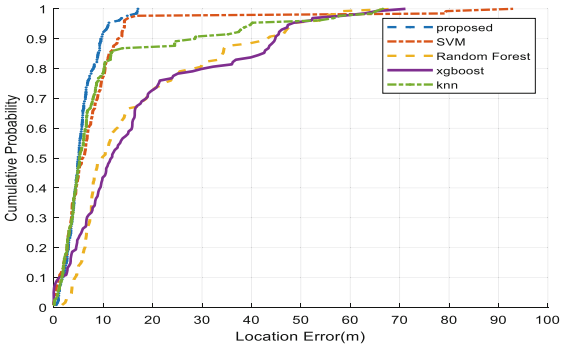

Fig. 8. CDF of location error to evaluate robustness in terms of hard ware, Training data and test data collected by Mi4_2 and S5 respectively

TIME STEP is 4, the performance on this data set is the best (4.31 m), and the training time gradually increases with the length of the TIME STEP.

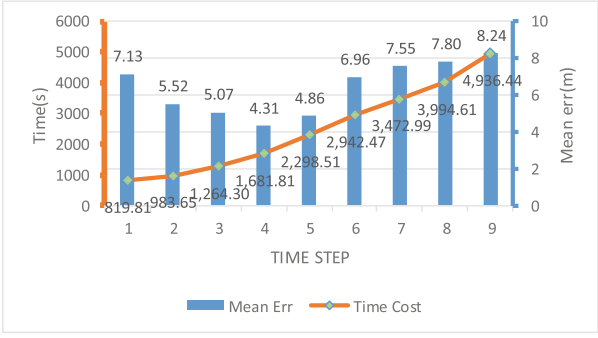

Fig. 9. The performance of proposed method with respective to value of TIME STEP

4.3.4 Comparison Scheme on Time

In this section, the train data and test data were collected in 2015 and 2017 respectively. Because data (in Fig. 10) was collected not only at different time but also by different devices, we just make the preliminary decision that the proposed method can relieve the RSS's fluctuation over time through the rough experiment and the literature [3]. From the Fig. 10, it is obvious that the proposed data also reach the lowest RMS, implying that proposed method can alleviate the RSS fluctuation over time.

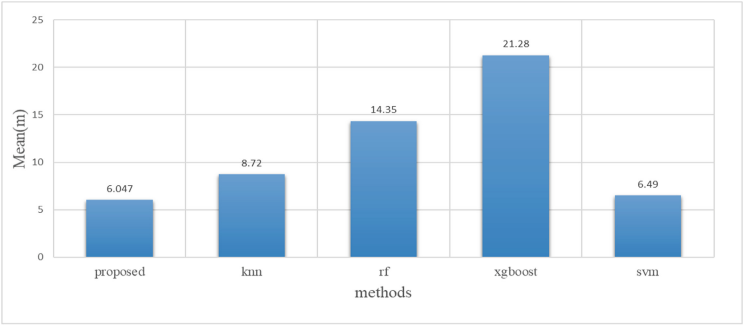

Fig. 10. The performance of methods on Time fluctuation

5 Conclusion

This work proposed an indoor positioning algorithm for sequence matching by fusing fingerprints and fingerprint spatial gradients with the encoder-decoder framework, which is a deep learning technique. The algorithm has effectively alleviated hardware heterogeneity and spatial ambiguities and improved the indoor positioning accuracy by 1.17 m, which is superior to the state-of-the-art algorithms proposed in the literature. As the data were collected on the same day, it is impossible to verify the fingerprint gradient fusion and the mitigation effect of time instability, which remains to be explored. In the future, we will combine our model with the Inertial Navigation technique to improve the robustness of the system when the wireless signal is weak or unreliable. We will also add the barometer to our deep learning framework to further determine the floor level while the user is using an elevator or a lift to achieve ubiquitous indoor positioning.

References

1. Wei, S., Wang, J., Zhao, Z.: LocTag: passive WiFi tag for robust indoor localization via smartphones. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE (2020)
2. Shi, L., et al.: 5G Internet of radio light positioning system for indoor broadcasting service. IEEE Trans. Broadcast. **66**(2), 534–544 (2020)

3. Wu, C., et al.: Gain without pain: accurate WiFi-based localization using fingerprint spatial gradient. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 2, pp. 1–19 (2017)
4. Mirowski, P., et al.: Probability kernel regression for WiFi localisation. J. Location Based Serv. **6**(2), 81–100 (2012)
5. Li, Y., et al.: Deep reinforcement learning (DRL): Another perspective for unsupervised wireless localization. IEEE Internet Things J. **7**(7), 6279–6287 (2019)
6. Niu, Q., et al.: DeepNavi: a deep signal-fusion framework for accurate and applicable indoor navigation. In: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 3, no. 3, pp. 1–24 (2019)
7. Hoang, M.T., et al.: Recurrent neural networks for accurate RSSI indoor localization. IEEE Internet Things J. **6**(6), 10639–10651 (2019)
8. Zheng, Y., Jiing'ao, X., Lina, Y.: Indoor localization: challenges and opportunities. J. Northwest Univ. (Nature Science Edition). **48**(2), 172–182 2018
9. Tokushige, H., Fossorier, M.P.C., Kasami, T.: A test pattern selection method for a joint bounded-distance and encoding-based decoding algorithm of binary codes [Transactions Letters]. In: IEEE Transactions on Communications, vol. 58, no. 6, pp. 1601–1604 (2010)
10. Olah, C.: Understanding LSTM Networks (2015)
11. Hinton, G.E., Ruslan, R.S.: Reducing the dimensionality of data with neural networks. Sci. **313**(5786), 504–507 (2006)
12. Zhuang, Y., et al.: Visible light positioning and navigation using noise measurement and mitigation. In: IEEE Transactions on Vehicular Technology, vol. 68, no. 11, pp. 11094–11106 (2019)
13. Han, J.B., Choi, L.: Large-scale indoor positioning using geomagnetic field with deep neural networks. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). IEEE (2019)
14. Rizk, H., Torki, M., Youssef, M.: CellinDeep: robust and accurate cellular-based indoor localization via deep learning. IEEE Sens. J. **19**(6), 2305–2312 (2018)
15. Bahl, P., Padmanabhan, V.N.: RADAR: an in-building RF-based user location and tracking system. In: Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), vol. 2. IEEE (2000)