Recurrent Neural Networks For Accurate RSSI Indoor Localization

Minh Tu Hoang, Brosnan Yuen, Xiaodai Dong, Tao Lu, Robert Westendorp, and Kishore Reddy

Abstract—[1] **This paper proposes recurrent neuron networks (RNNs) for a fingerprinting indoor localization using WiFi. Instead of locating user's position one at a time as in the cases of conventional algorithms, our RNN solution aims at trajectory positioning and takes into account the relation among the received signal strength indicator (RSSI) measurements in a trajectory. Furthermore, a weighted average filter is proposed for both input RSSI data and sequential output locations to enhance the accuracy among the temporal fluctuations of RSSI. The results using different types of RNN including vanilla RNN, long short-term memory (LSTM), gated recurrent unit (GRU) and bidirectional LSTM (BiLSTM) are presented. On-site experiments demonstrate that the proposed structure achieves an average localization error of 0.75 m with 80% of the errors under 1 m, which outperforms the conventional KNN algorithms and probabilistic algorithms by approximately 30% under the same test environment.**

Index Terms- **Received signal strength indicator (RSSI), WiFi indoor localization, recurrent neuron network (RNN), long short-term memory (LSTM), fingerprint-based localization.**

I. INTRODUCTION

The motivation of this paper is to locate a walking human using the WiFi signals of the carried smartphone. In general, there are two main groups in WiFi indoor localization including model based and fingerprinting based methods. The former one utilizes the propagation model of wireless signals in forms of the received signal strength (RSS), the time of flight (TOF) and/or angle of arrival (AOA) [1] to estimate the location of the target, while the latter considers the physically measurable properties of WiFi as fingerprints or signatures for each discrete spatial point to discriminate between locations. Due to the wide fluctuation of WiFi signals [2] in an indoor environment, the exact propagation model is difficult to obtain, which makes fingerprinting approach more practicable for the WiFi based localization.

In WiFi fingerprint, received signal strength indicator (RSSI) is widely used as a feature in localization because RSSI can be obtained easily from most WiFi receivers such as mobile phones, tablets, laptops, etc. [3]. However, RSSI has two drawbacks: instability due to fading and multipath effects and device heterogeneity, i.e., different devices have different RSSIs even at the same position [3]. In order to mitigate those problems, channel state information (CSI) is adopted to provide richer information from multiple antennas and multiple subcarriers [3], [4]. Although CSI is a more reliable fingerprint to improve the localization accuracy, it is only available with the specific wireless network interface cards (NIC), e.g., Intel WiFi Link 5300 MIMO NIC, Atheros AR9390 or Atheros AR9580 chipset [5]. Therefore, RSSI is still a popular choice in practical scenarios.

Among WiFi RSSI fingerprinting indoor localization approaches, the probabilistic method is based on statistical inference between the target signal measurement and stored fingerprints using Bayes rule [6]. The probability density function (PDF) of the RSSIs is assumed as empirical parametric distributions (e.g., Gaussian, double-peak Gaussian, lognormal [7], [8]), which may not be necessarily true in practical situations. In order to achieve better performance, non-parametric methods [9], [10] make no assumption on the PDF of RSSI but require a large amount of data at each reference point (RP) to form the smooth and accurate PDF. Beside the probabilistic approach, the deterministic methods use a similarity metric to differentiate the measured signal and the fingerprint data in the dataset to locate the user's position [11]. The simplest type of this approach is K nearest neighbors (KNN) [12]–[14] which determines the user location by calculating and ranking the fingerprint distance measured at the unknown point and the reference locations in the database. Besides, support vector machine (SVM) [15], [16] provides a direct mapping from RSSI values collected at the mobile devices to the estimated locations through nonlinear regression by supervised classification technique [17]. Despite their low complexity, the accuracy of both of those methods are unstable due to the wide fluctuation of WiFi RSSI [13]–[15]. Different from those previous algorithms, artificial neural network (ANN) [3], [18] estimates location nonlinearly from the input by a chosen activation function and adjustable weightings. In indoor environments, because the relationship between the RSSI values and the user's locations is nonlinear, it is difficult to formulate a closed form solution [17]. ANN is a suitable and reliable solution for its ability to approximate high dimension and highly nonlinear models [3]. Recently, several ANN localization solutions such as multilayer perceptron [19], roburst extreme learning machine (RELM) [20], multi-layer neural network (MLNN) [21], convolutional neural network (CNN) [22], etc., have been proposed.

Although having been extensively investigated in literature, all of the above algorithms are still facing the following challenges [23]:

1 Spatial ambiguity: Some physically distant locations may have similar fingerprints or similar fingerprint distances compared with the current location.

2 RSSI instability: the observed fingerprint of a location

in the testing phase may not match that collected in the training phase because of the wide fluctuation of WiFi signals [2].

3 RSSI short collecting time per location: due to the mobile nature of the locating target, the RSSI sampling at each specific location in the testing phase is typically short. Therefore, only a few number of RSSI readings can be collected, which severely impairs the localization accuracy.

To address these challenges above, this paper focuses on recurrent neural network (RNN) which exploits the sequential RSSI measurements and the trajectory information to determine the user's location. The idea of exploiting the measurements in previous time steps to locate the current location was adopted before in the research of Kalman filter [24]–[27] and soft range limited K-nearest neighbors (SRL-KNN) [23]. Kalman filter estimates the most likely current location based on prior measurements and Gaussian noise with linear motion dynamics assumptions. However, in real scenarios, those assumptions are not necessarily valid [2]. In comparison, SRL-KNN does not require the above assumptions but needs the constraints about the bounded speed of the indoor users, e.g., from 0.4 to 2 m/s. If the users change their speed beyond the assumed maximum speed, the localization accuracy of SRL-KNN will be affected severely. In contrast to those approaches, the proposed RNN model does not require either the assumption of Gaussian noise distribution or the user's bounded speed. It makes the prediction by only adjusting the internal weights through training phase. Furthermore, in order to overcome the RSSI instability, the weighted average filter is applied in both training and testing RSSI measurements. The performance is tested in different types of RNN including vanilla RNN, long short term memory (LSTM) [28], gated recurrent unit (GRU) [29] and bi-directional LSTM (BiLSTM) [30].

The rest of the paper is organized as follows. Section II introduces related works on neural network and RNN for indoor localization, followed by details of the proposed RNN model in Section III. Section IV reports the experimental set-up and results for the performance evaluation. Finally, Section V concludes this paper.

II. RELATED WORK

The first research on neural network for indoor localization was reported by Battiti *et al.* [15], [32]. In that work, a multi-layer perceptron network (MLP) with 3 layers and 16 neurons was implemented to nonlinearly map the output (coordinate) from the input (RSSI). There are 207 reference points for training and 50 random points for validation. The accuracy of this system is 2.82 m ± 0.11 m, which is comparable with that of simple K nearest neighbours (KNN) algorithm [15]. The refinement is proposed in discriminant-adaptive neural network (DANN) [18], which inserts discriminative components (DCs) layer to extract useful information from the inputs. The experiment shows that DANN improves the probability of the localization error below 2.5 m by 17% over the conventional KNN RADAR [12].

In order to achieve better performance, multi-layer feed-forward neural network (MLNN) [21] tries to transforms and denoises the raw RSSI signals before classifying the labels
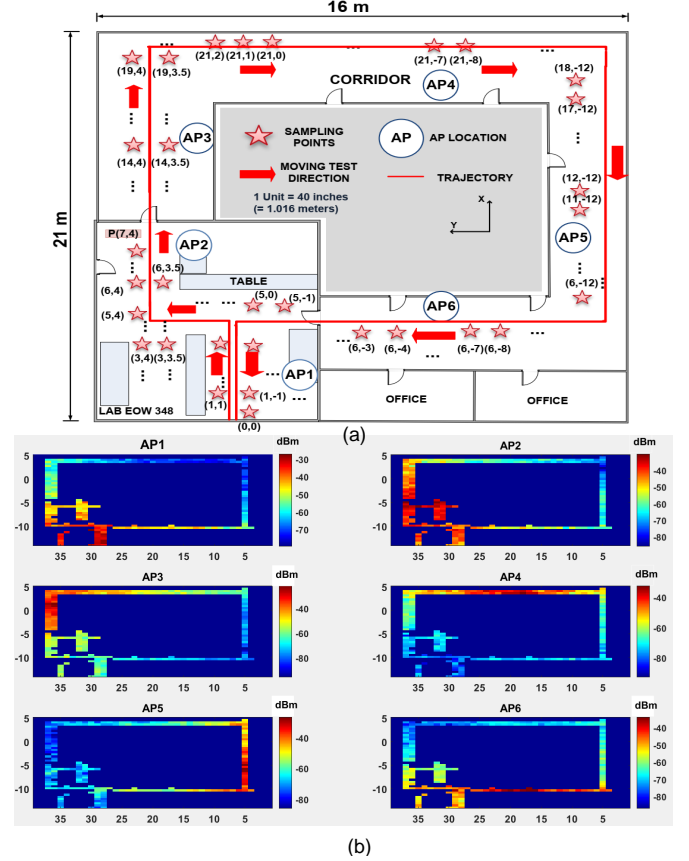

Fig. 1.   (a) Floor map of the test site. The solid red line is the mobile user's walking trajectory with red arrows pointing toward walking direction. (b) Heat map of the RSSI strength from 6 APs used in our localization scheme.

of the unknown locations. Based on the experiment, although MLNN has an acceptable accuracy, the localization errors grow rapidly when the square grid size becomes smaller and creates more location ambiguity. Therefore, a boosting method is required to add to MLNN to tune it in the case of misclassified data.

As all of those above methods are time-consuming in training, robust extreme learning machine (RELM) [20] is proposed to increase the training speed with the enhanced performance by overcoming the fluctuation of RSSI. RELM consists of a generalized single-hidden layer feedforward neural network with the addition of the random hidden nodes initialization and kernelized formulations. The experiment illustrates that the training speed of RELM is over 100 times faster than the conventional support vector regression (SVR) [33]. Furthermore, RELM outperforms SVR in accuracy by around 40%.

Convolutional neural network was adopted for indoor WiFi localization in ConFi [3]. The channel state information for different subcarriers at different time is arranged into a matrix, which is similar to an image. The three layers CNN is trained using these CSI feature images collected at a number of reference points. The localization results is the weighted centroid of RPs with high output value. The experiment illustrates that ConFi outperforms conventional KNN RADAR [12] by 66.9% in term of mean localization error. However, as mentioned in Section I, it is not easy to get CSI because only some specific

TABLE I. COMPARISONS OF INDOOR LOCALIZATIONS USING MACHINE LEARNING TECHNIQUES

Method	Feature	Number of APs	Number of RPs	Number of Testing Points	Grid Size	Accuracy
MLP [19]	RSSI	6	207	50	1.7 m	2.8 m $\pm$ 0.1 m
DANN [18]	RSSI	15	45	46	2 m	2.2 m $\pm$ 2.0 m
RELM [20]	RSSI	8	30	10	3.5 m	3.7 m $\pm$ 3.4 m
MLNN [21]	RSSI	9	20	20	1.5 m	1.1 m $\pm$ 1.2 m
ConFi [3]	CSI	1	64	10	1.5 - 2 m	1.3 m $\pm$ 0.9 m
Geomagnetic RNN [31]	Magnetic Information	-	629	5% of RPs	0.57 m	1.1 m

wireless network interface cards, e.g, Intel WiFi Link 5300 MIMO NIC, Atheros AR9390 or Atheros AR9580 chipset can obtain that information.

Recently, recurrent neural network is used to locate a user's position in an indoor environment. [34] proposed a simple RNN with 1 time step and 2 hidden layers. RNN classifies 42 RPs based on the RSSI readings from 177 APs. The classification accuracy is 82.47%. A more efficient RNN is published in [31]. In this work, the RNN has 200 neurons uses mean squared error (MSE) loss function and has 20 time steps trace as the input for the network. In that work, geomagnetic data is used as fingerprints instead of the wireless data. A million traces of various pedestrian walking patterns are generated with 95% of them for training and 5% for validation. The achieved localization errors range from 0.441 to 3.874 m with the average error being 1.062 m. In addition to the geomagnetic data, the light intensity is also utilized in the deep LSTM network [35] to estimate the indoor location of the target mobile device. Their 2 layers LSTM exploits temporal information from bimodal fingerprints, i.e., magnetic field and light intensity data, through recursively mapping the input sequence to the label of output locations. The accuracy is reported as 82% of the test locations with location errors around 2 m and the maximum error being around 3.7 m.

Table I summarizes the experimental set-up and the results from some of the mentioned neural network methods above. These methods provide acceptable accuracy from 1 m to 3 m but none of them sufficiently investigated the 3 problems of using RSSI fingerprints as stated in Section I, i.e., spatial ambiguity, RSSI instability, RSSI short collecting time per location. Furthermore, according to the authors' knowledge, there is no comprehensive RNN solution using WiFi RSSI fingerprinting with appropriate analysis, details and comparisons. Therefore, we propose a detailed RNN solution for fingerprint indoor localization using WiFi to solve the above three challenges. Besides, different types of RNN including vanilla RNN, long short-term memory, gated recurrent unit, and bidirectional LSTM and their structure with all of the important parameters will be analysed in detail. Our localization results of RNN are compared not only with the other neural network methods [19], [21] but also some conventional methods including KNN [12], Kernel method [9] and Kalman filter [24] using 2 different databases, one in house measurements and one publicly available data source.

III. SYSTEM MODEL

A. Localization Scene

The fingerprinting localization system is generally divided into two phases: a training phase (offline phase) and a testing phase (online phase). In the training phase, RSSI readings at each predefined reference point (RP) location, are collected and stored to a database as fingerprints. Assuming that the area of interest has P APs and M RPs, we consider each RP i at its physical location $l_i(x_i, y_i)$ having a corresponding fingerprint vector $f_i = \{F_1^i, F_2^i, ..., F_N^i\}$, where N is the number of available features and $F_j^i (1 \leq j \leq N)$ is the j-th feature at RP i. In the testing phase, each unknown location of the user, denoted as a testing point, is determined by the localization algorithm. During the training phase, a set of RSSI readings (S_1 scans) are collected at a single RP while in the testing phase, and only a small number of RSSI readings (S_2 scans), e.g., $S_2 = 1$ or $S_2 = 2$, is available as the user is expected to frequently moving in practical scenario. Fig. 1(a) illustrates the localization map with 6 APs, 365 RPs and 175 testing points, while Fig. 1(b) shows the heat map of these 6 APs.

B. Recurrent Neural Network model

A recurrent neural network is a class of artificial neural network where the output results depend not only on the current input value but also on the historical data [31]. RNN is often used in situations where data has a sequential correlation. In the case of indoor localization, the current location of the user is correlated to its previous locations as the user can only move along a continuous trajectory. Therefore, RNN is appropriate to exploit the sequential RSSI measurements and the trajectory information to enhance the accuracy of the localization. Up to date, there are several RNN variants because the simplest one, vanilla RNN [36] has limited applications due to vanishing gradient during the training phase. To mitigate that effect, long short term memory [28] creates an internal memory state which adds the forget gate to control the time dependence and effects of the previous inputs. Gated recurrent unit [29] shares a similar idea to that of LSTM but reduces from 3 gates of LSTM, i.e., forget, update and output gate, to only 2 gates, i.e., update and reset gate. Bidirectional LSTM [30] is an extension of the traditional LSTM, which not only utilizes all available input information from the past but also from the future of a specific time frame. In the following section we will describe several important parameters in RNNs which includes memory length, input and output feature and loss function.

1) Memory Length: The proposed RNN architecture is trained by the data from consecutive locations (trajectories) to exploit the time correlation between them. Each location in a trajectory appears in one different time step. The length of a trajectory (the number of time steps) defines the memory length T (Fig. 2). Because all of the weights, hidden state values will be saved in every time step in a training trajectory [36], the number of time steps T affects significantly the performance of RNN. Larger T will reveal more information from the past but accumulate more errors on prediction. The appropriate T will be chosen by the experiment in Section IV.

2) Input and Output Feature: Fig. 2 illustrates 5 different possible training models of the proposed RNN architecture.
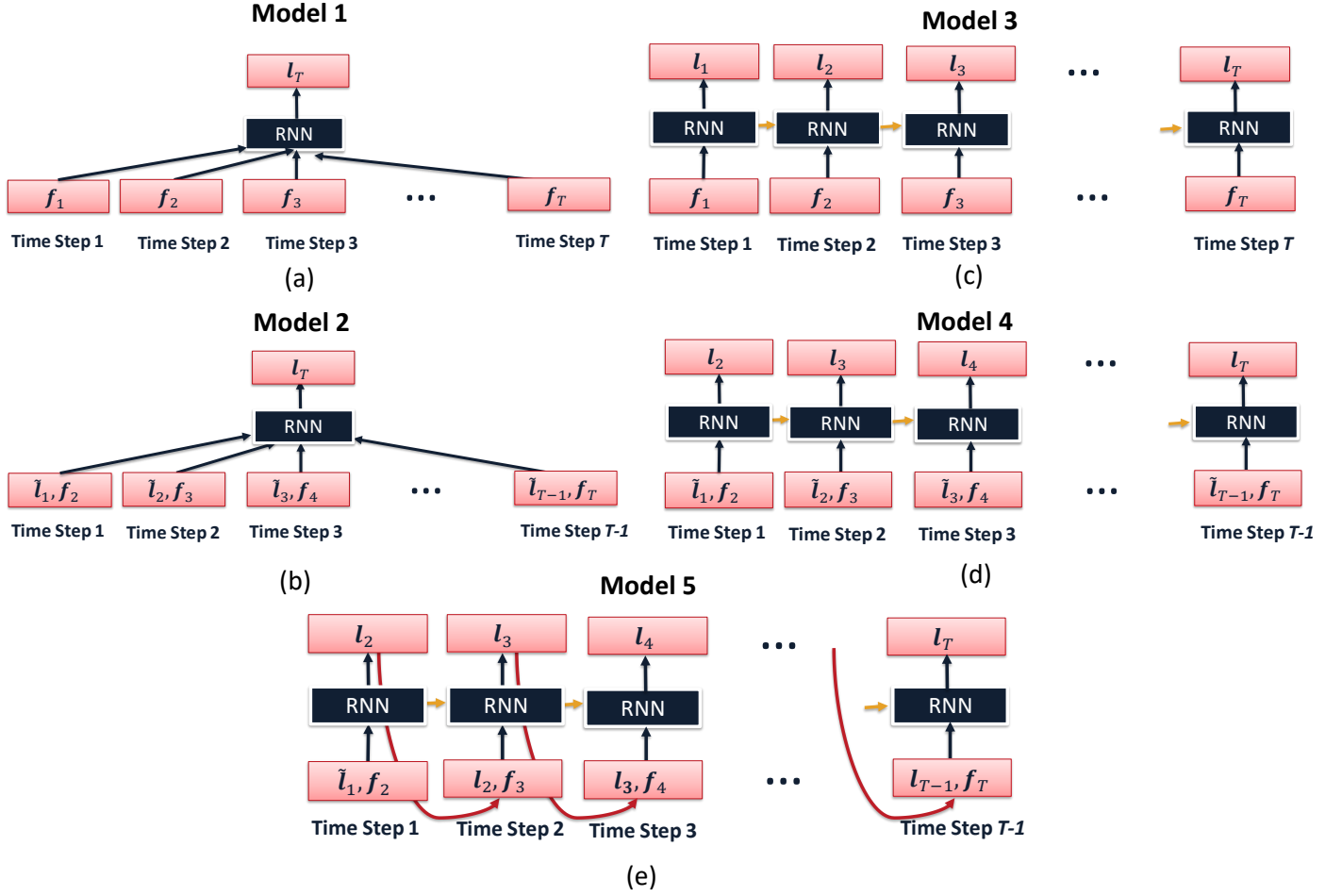
Fig. 2. Different possible models of the proposed RNN.

The input feature used by model 1 and 3 are the RSSI information ($\boldsymbol{f}_i$) from T time steps. In contrasts, model 2, 4 and 5 use both the previous locations and RSSI information. Specifically, in model 2 and 4, the location used in the training is the actual one ($\tilde{\boldsymbol{l}}_i$) recorded in the dataset. In model 5, the location value is the predicted location ($\boldsymbol{l}_i$) from the previous time step. The output feature of model 1 and 2 are the location of the time step T ($\boldsymbol{l}_T$). In general, those models base on the information of the previous $T-1$ steps to predict the location of step T. On the other hand, model 3, 4 and 5 have the output locations $\boldsymbol{l}$ in every time step. All of these 5 models will be tested and compared in Section IV.

3) Loss Function: The objective of RNN training is minimizing the loss function $\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}})$ which penalizes the Euclidean distance between the output $\boldsymbol{l}$ and the target $\tilde{\boldsymbol{l}}$. Then the backpropagation algorithm uses the chain rule to calculate the derivative of the loss function $\mathcal{L}$ and adjusts the network weights by gradient descent [36]. The distance is chosen as root mean square error (RMSE). Model 1 and 2 with one output in each trajectory training use the loss function given by

$$\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}}) = ||\boldsymbol{l}_T - \tilde{\boldsymbol{l}}_T||_2. \tag{1}$$

Model 3, 4 and 5 with T outputs for each trajectory training use the loss function given by

$$\mathcal{L}(\boldsymbol{l}, \tilde{\boldsymbol{l}}) = \frac{\sum_{i=1}^{T} ||\boldsymbol{l}_i - \tilde{\boldsymbol{l}}_i||_2}{T}. \tag{2}$$

C. Localization Process

The architecture of the proposed RNN system is presented in Fig. 3(a). The RSSI data for both training and testing will be collected with the support of an autonomous robot. The 3-wheel robot (Fig. 3(a)) has multiple sensors including a wheel odometer, an inertial measurement unit (IMU), a LIDAR, sonar sensors and a color and depth (RGB-D) camera. The robot can navigate to a target location to collect WiFi fingerprints autonomously. The localization accuracy of the robot is 0.07 m ± 0.02 m. A mobile device (Google Nexus 4 running Android 4.4) mounted on a 3-wheel robot was sent to target locations to collect fingerprints. At each location, 100 instantaneous RSSI measurements ($S_1 = 100$) were collected for offline training. On the other hand, in the online testing, only a small number of RSSI readings 1 or 2 ($S_2 = 1$ or 2) is available. There are 6 APs and 5 of them provide 2 distinct MAC address for 2.4 GHz and 5 GHz communications channels respectively. Equivalently, in every scan, 11 RSSI readings from those 6
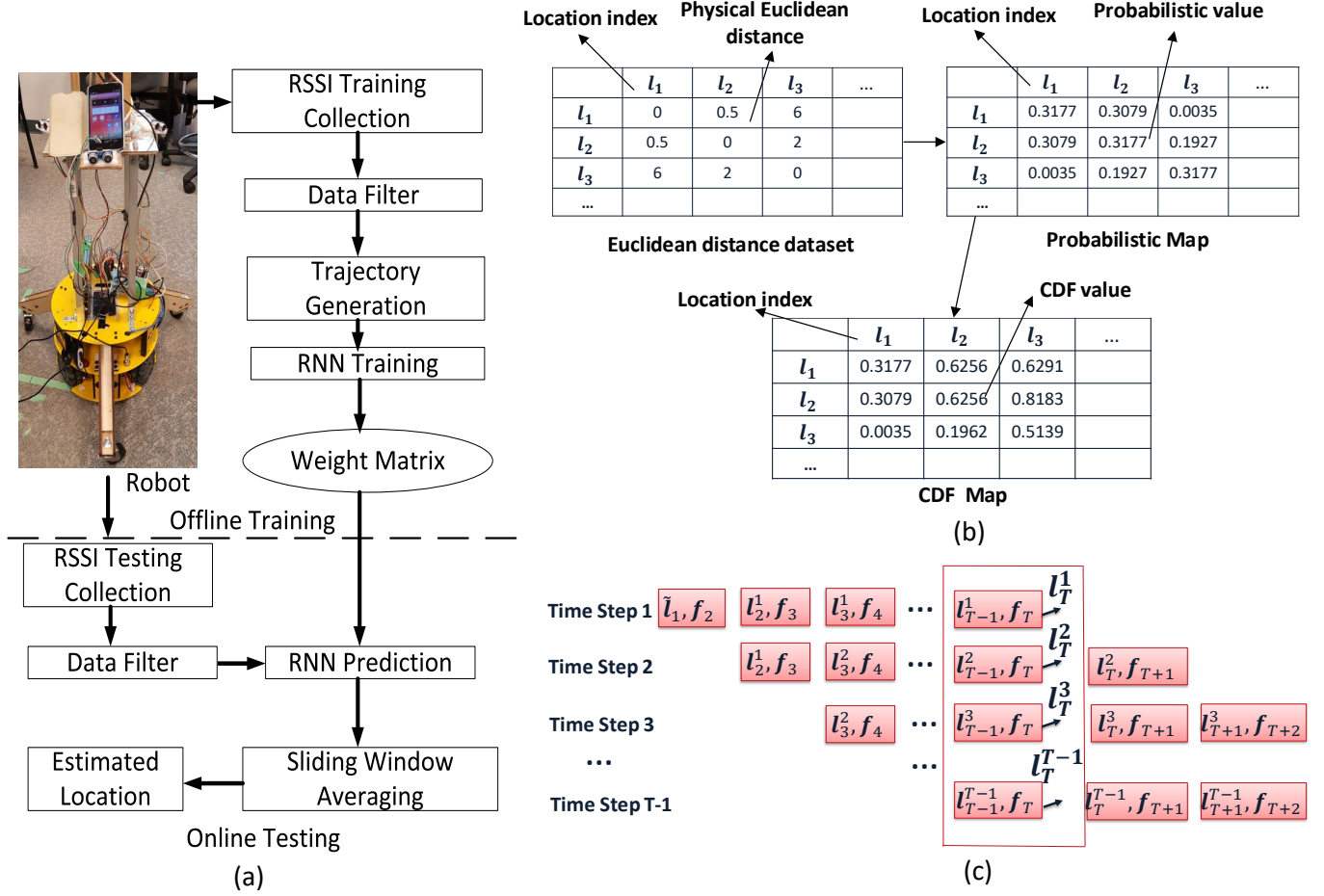
Fig. 3. (a) Localization process of the proposed RNN system. (b) Trajectory generation process. (c) Sliding window averaging in online testing phase.

Euclidean distance dataset — Location index / Physical Euclidean distance

	l_1	l_2	l_3	...
l_1	0	0.5	6	
l_2	0.5	0	2	
l_3	6	2	0	
...				

Probabilistic Map — Location index / Probabilistic value

	l_1	l_2	l_3	...
l_1	0.3177	0.3079	0.0035	
l_2	0.3079	0.3177	0.1927	
l_3	0.0035	0.1927	0.3177	
...				

CDF Map — Location index / CDF value

	l_1	l_2	l_3	...
l_1	0.3177	0.6256	0.6291	
l_2	0.3079	0.6256	0.8183	
l_3	0.0035	0.1962	0.5139	
...				

APs can be collected. Details of the process will be described as follows.

1) Data Filter: The RSSI collected by a client device often experiences substantial fluctuations due to dynamically changing environments such as human blocking and movements, interference from other equipment and devices, receiver antenna orientation, etc., [37], [38]. In our experiment, the mobile device was put on the location $P(7, 4)$ as shown in Fig. 1(a). The experiment was conducted in busy hours when many students (up to 10) used WiFi and walked around the lab. A maximum RSSI standard deviation of 5.5 dB was recorded over 100 consecutive RSSI readings, and 5% of the measurements (5 readings) could not be detected (missing values). In those cases, the client device missed the beacon frame packets sent by the router. In order to filter out those outliers and make the measurement valid, we adopt the iterative recursive weighted average filter [39]. Based on the transfer function in Fig. 4, the weighted average filter has the form of a low pass filter. The effectiveness of the filter will be studied in Section IV.

2) Trajectory Generation: The proposed RNN system is trained by the consecutive RSSI measurements and locations in the trajectory. Therefore, we generate random training
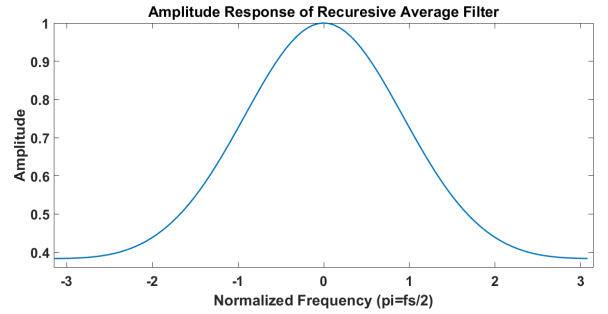

Fig. 4. Weighted average filter transfer function.

trajectories based on the fact that the moving speed of the user in an indoor environment is bounded. At each location in the trajectory, we randomly choose one out of 100 RSSI stored in the database as the RSSI associated with the location. This approximates well the user's random walk property and helps to reduce the spatial ambiguity problem. Fig. 3(b) illustrates the trajectory generation process. Firstly, the physical Euclidean distance between each location and the rest of the database is calculated to form a Euclidean distance dataset. Secondly, based on that, the probabilistic map

will be generated to represent the probability of a location ($P(\boldsymbol{l}_i)$) that will become the next location of the user in the trajectory. Since the moving speed of an indoor user is limited, the locations which are near to the previous locations should have higher probability to become the next location in the trajectory than the further locations. The user location will be updated in every consecutive sampling time interval Δt. Therefore, the maximum distance which the user can move in Δt is $\sigma = v_{max} \times \Delta t$, where the maximum speed v_{max} is chosen to be larger than the human indoor normal speed (from 0.4 m/s to 2 m/s [40], [41]). The normalized probability $P(\boldsymbol{l}_i)$ is calculated as follows.

$$P(\boldsymbol{l}_i) = \frac{1}{2\sigma^2(1 - e^{\frac{d_{max}^2}{2\sigma^2}})} \exp(-\frac{(x_i - x_{pre})^2 + (y_i - y_{pre})^2}{2\sigma^2}) \tag{3}$$

where (x_{pre}, y_{pre}) is the most recent previous location of $\boldsymbol{l}_i$, d_{max} is the maximum distance between the considered location $\boldsymbol{l}_i$ and the furthest location in the interested area. (3) has the form of a Gaussian distribution with the mean being the previous location and the standard deviation being σ. The sum of all $P(\boldsymbol{l})$ in the interested area being 1 is proved in the Appendix. All of the locations having the same physical distance with $\boldsymbol{l}_i$ will get the same probability to be chosen as the next point on the trajectory. From the probabilistic map, a CDF map is built by summing the $P(\boldsymbol{l}_i)$ for each location $\boldsymbol{l}_i$. Finally, in order to get the next location $\boldsymbol{l}_j$ of a any location $\boldsymbol{l}_i$ in the trajectory, a random number R ($0 < R < 1$) is picked. $\boldsymbol{l}_j$ is the location which has the value in CDF map being closest with R.

3) Sliding Window Averaging: Model 3, 4 and 5 have the output locations appearing in every time step (Subsection III-B2). Therefore, in the online testing phase, the output location $\boldsymbol{l}_T$ will be appeared in several time steps. Fig. 3(c) illustrates the sliding window averaging for location $\boldsymbol{l}_T$ in model 5. $\boldsymbol{l}_i^j$ is the output location $\boldsymbol{l}_i$ of time step j. At the time step $T-1$, we have a set of output location $(\boldsymbol{l}_T^1, \boldsymbol{l}_T^2, ..., \boldsymbol{l}_T^{T-1})$ from $T-1$ previous steps. The final output result $\boldsymbol{l}_T$ will be the average of that set:

$$\boldsymbol{l}_T = \frac{\sum_{j=1}^{T-1} \boldsymbol{l}_T^j}{T-1}. \tag{4}$$

IV. EXPERIMENT AND ANALYSIS

A. Experimental Setup

All experiments have been carried out on the third floor of Engineering Office Wing (EOW), University of Victoria, BC, Canada. The dimension of the area is 21 m by 16 m. It also has 3 long corridors as shown in Fig. 1(a). There are 365 RPs and 175 random testing locations. The initial position of the user in the testing trajectory is known. In training, at each location, 100 instantaneous RSSI measurements ($S_1 = 100$) were collected. Among them, only one randomly picked measurement is used for each location in each testing trajectory. In the online phase, the robot moved along a pre-defined route as shown in Fig. 1(a) with an average speed around 0.6 m/s. RSSI readings were collected continuously by the phone and were transmitted to a server in real time. The server analyzed the data to locate the user's position.

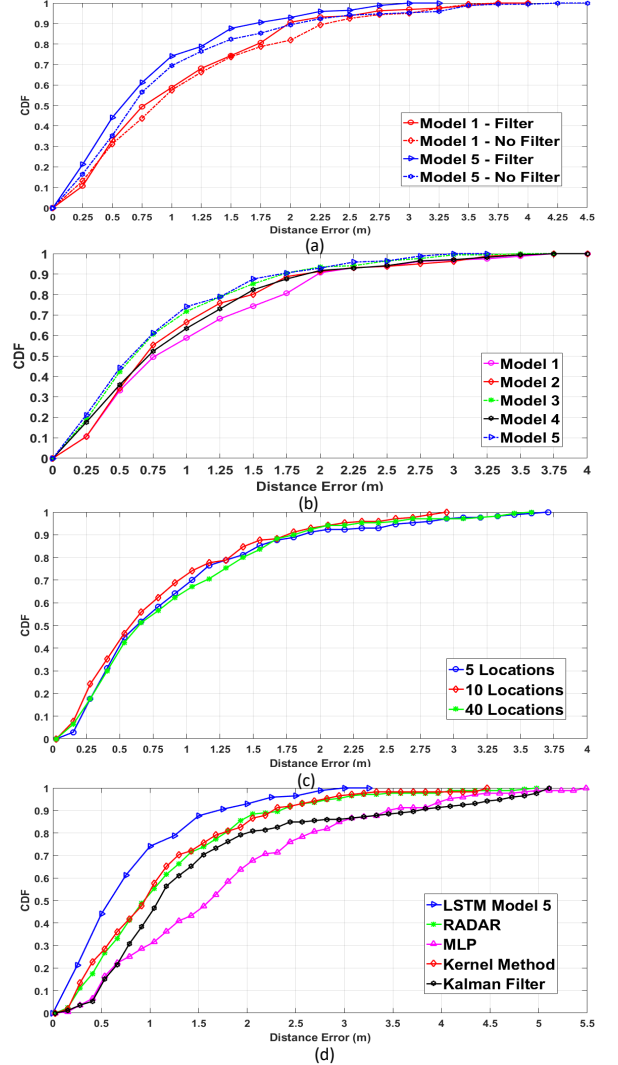| Category	Value
RNN type	LSTM
Memory length	10
Model	5
Loss function	RMSE
Hidden layer (HL)	2
Number of neurons for each HL	100
Dropout	0.2
Optimizer	Adam
σ	2 m
Δ_t	1 s
d_{max}	2 m

Fig. 5. The CDF of the localization error of (a) Filter and no filter cases (b) 5 different RNN models (c) Different memory length in RNN structure (d) LSTM model 5 and the other methods in literature.

The initial setup for the proposed RNN system follows the parameters presented in Table II. All of the results below are presented after 10-fold tests with each subset test including $20,000$ random training trajectories.

B. Filter Comparison

Fig. 5(a) compares the cumulative distribution function (CDF) of localization errors between the proposed RNN mod-
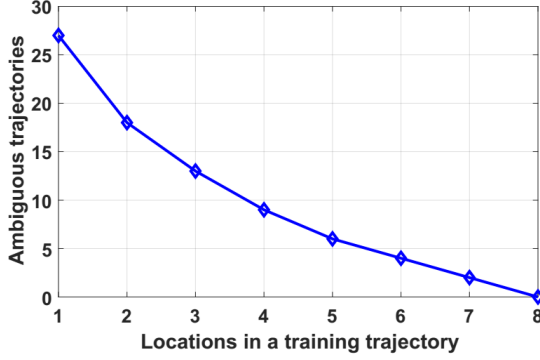
Fig. 6.   Average number of ambiguous trajectories with different number of locations in a training trajectory

els (i.e., model 1 and 5) with 2 different cases: using or not using weighted average filter (Subsection III-C1) in both the dataset and testing data. For RNN model 5 (blue line), the filter helps to decrease the maximum error from 4.5 m to 3.25 m. Besides, 80% of the error is within 1.25 m with the filter while no filter case is 1.5 m. In RNN model 1, the filter also illustrates better performance with 80% of the error being within 1.75 m compared with 2 m of the no filter case. In total, the weighted average filter provides good effects to the performance of the proposed RNN algorithms. Therefore, all of the results presented in the rest of the article are all using the filter for both database and testing data.

C. Model Comparison

Table III illustrates the average errors between 5 different proposed LSTM models (Subsection III-B2). The best performance is 0.75 m with the standard deviation 0.64 m in the case of LSTM model 5. LSTM model 3 provides the second best performance with the average error of 0.80 m $\pm$ 0.67 m. The worst accuracy belongs to LSTM model 1 being 1.05 m $\pm$ 0.78 m. Fig. 5(a) compares the CDF errors between these 5 models. Model 3 and 5 consistently show the dominating accuracy with 80% of the errors within 1.2 m, compared with 1.7 m of model 1. Besides, the maximum error of model 3 is 3.25 m which is lower than 4 m in the case of model 1. In the rest of the paper, LSTM model 5 is chosen for further performance study.

D. Memory Length Comparison

Fig. 5(c) shows the results of LSTM model 5 with different memory length (Subsection III-B1), i.e., a training trajectory has 5, 10 or 40 locations. The performance of all 3 cases is comparable. The 10 time step training trajectory has a slightly

better accuracy with the maximum error being only 2.9 m, compared with 3.5 m and 3.75 m in the case of 40 time steps and 5 time steps respectively.

The theoretical explanation is as follows. A location l_j is defined as an ambiguous point of l_i if their physical distance is larger than the grid size but their two vectors f_i and f_j have high Pearson correlation coefficient above the correlation threshold. Besides, two locations are defined as physical neighbours if the physical distance between them is less than or equal to the grid size. The correlation threshold is chosen based on the average correlation coefficients between l_i and all of its physical nearest neighbours, i.e., approximately 0.9 in our database. Then all non-nearest-neighbour locations whose correlation coefficient above this threshold are considered as ambiguous points. Pearson correlation coefficient $\rho(f_i, f_j)$ between f_i and f_j can be calculated as follows

$$\rho(f_i, f_j) = \frac{1}{N-1} \sum_{n=1}^{N} \left(\frac{F_n^i - \mu_i}{\delta_i}\right)\left(\frac{F_n^j - \mu_j}{\delta_j}\right)$$

where μ_i, μ_j are the mean of f_i and f_j respectively, δ_i, δ_j are the standard deviation of f_i and f_j respectively. Similar to the definition of the ambiguous location, 2 trajectories are defined as ambiguous if they include different locations but the combinations of their fingerprints have a high Pearson correlation coefficient. Fig. 6 demonstrates the advantage of the proposed LSTM model which exploits the sequential trajectory locations compared with the conventional single point prediction. In the case of single point prediction (1 location in a training trajectory), the average number of ambiguous locations in our database are 27. If we increase the number of locations in a trajectory for training, the number of ambiguous trajectories decrease significantly. If a training trajectory has more than 8 locations, there will be no ambiguity in our database. Therefore, our memory length configured before as 10 locations is reasonable to remove all of the ambiguity.

E. Number of Neurons Comparison

Table IV compares the performance of LSTM model 5 with different number of neurons in each hidden layer. Note that all of the results in the above sections are based on the initial network setting with 2 hidden layers and 100 neurons per layer (Table II). If the number of neurons in each hidden layer increase to 250 and 500, the accuracy is mostly similar to the case of 100 with the average error being around 5% different, i.e., 0.80 ± 0.65 m and 0.82 ± 0.65 m respectively. On the other hand, if the number of neurons drops to 50 neurons per layer, the average error increase to 0.89 ± 0.70 m. In the case of using 100 neurons in hidden layer 1 and only 50 neurons in hidden layer 2, the performance is slightly worse than the
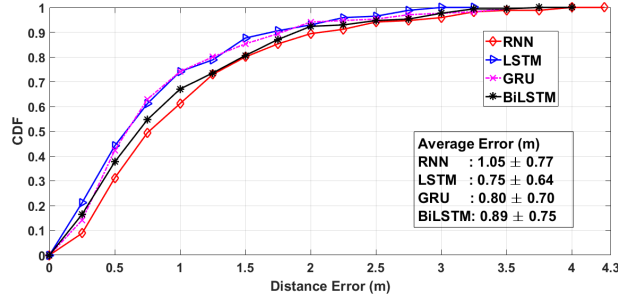
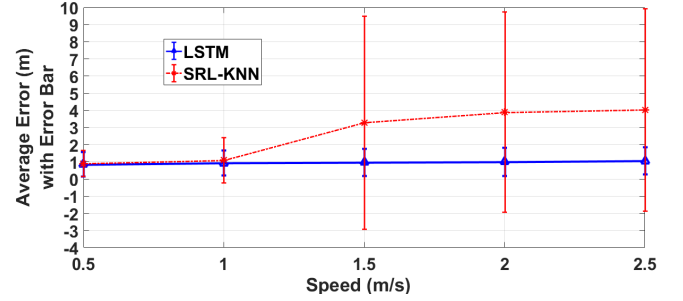Fig. 7. The CDF of the localization error of RNN, LSTM, GRU and BiLSTM

Fig. 8. Average localization errors with the error bars of LSTM and SRL-KNN in changing speed scenarios

TABLE V. AVERAGE LOCALIZATION ERRORS OF UJIINDOORLOC DATABASE

	LSTM	RADAR [12]	MLNN [21]
Building 0 (m)	4.5 ± 2.7	7.9 ± 4.9	7.6 ± 4.2
Building 1 (m)	4.0 ± 3.8	8.2 ± 4.9	7.5 ± 3.3
All buildings (m)	4.2 ± 3.2	8.1 ± 4.9	7.5 ± 3.8

case of using both 100 neurons in each hidden layer. In total, 2 hidden layers and 100 neurons per layer are the optimal parameters for the LSTM network.

F. Literature Comparison

Fig. 5(d) compares the proposed LSTM model 5 with the feed-forward neural network MLP [19] and the other conventional methods including KNN-RADAR [12], probabilistic Kernel Method [9] and Kalman Filter [24]. LSTM clearly outperforms MLP with the maximum error of 3.4 m compared with 5.5 m of MLP. 80% of LSTM model errors are within 1.2 m, which is much lower than 2.7 m of MLP. The maximum errors of conventional methods such as RADAR, Kalman filter and Kernel method are more significant 4.8 m, 5.0 m and 4.50 m, respectively. Besides, 80% of the errors of those methods are all within around 2 m, 1.7 times higher than the proposed LSTM model. Table III lists the average errors between LSTM models and the other mentioned methods. Clearly, the accuracy of LSTM model 5 with 0.75 m dominates the other conventional feed-forward neural networks with 1.65 m of MLNN [21] and 1.72 m of MLP [19].

G. RNNs Comparison

Fig. 7 compares the performance between vanilla RNN [31], LSTM [28], GRU [29] and BiLSTM [30]. All of the settings follow Table II. Although the gap between these systems are close, LSTM still consistently has the best performance with the average error is 0.75 ± 0.64 m compared with 0.80 ± 0.70 m, 0.89 m and 1.05 m of GRU, BiLSTM and RNN, respectively. While 80% of the errors of RNN and BiLSTM are all within 1.5 m, the ones of LSTM and GRU are only within 1.2 m.

H. Further Discussion

As mentioned in Section I, the proposed LSTM model can address these three challenges of WiFi indoor localization. Subsection IV-D has demonstrated that LSTM adopts a sequential measurements from several locations in the trajectory and decreases significantly the spatial ambiguity (first challenge). In addition, RSSI instability (second challenge) and RSSI short collecting time per location (third challenge) create the diverse values of RSSI readings in one location, which can lead to more locations having similar fingerprint distances. The proposed LSTM model can effectively remove the number of false locations which are far from the previous points based on the series of the previous measurements and predictions.

Therefore, the adverse effects of the second and third challenge can be mitigated.

Furthermore, some conventional short-memory methods such as Kalman filter [24], SRL-KNN [23] have the constraints about the speed of the users. If the users change their speed rapidly, the localization accuracy of those methods will be affected severely. In contrast, the proposed LSTM network is trained with random trajectories as described in Subsection III-C2 with no strict constraints about the speed of the users, so it can handle the changing speed scenarios. Fig. 8 illustrates the average errors of the proposed LSTM model 5 and SRL-KNN using RSSI mean database with parameter $\sigma = 2$ m [23]. The number of testing points are 344 locations following the backward and forward trajectory like Fig. 1. The maximum speed is the instant speed of the robot between 2 random consecutive testing locations in a sampling time interval Δt. The number of locations having the maximum speed are 50% of the total testing points (172 locations). The rest of the locations have a random speed smaller than the maximum speed. When the maximum speed increases from 0.5 m/s to 2.5 m/s, the average errors of LSTM model stays stable around 0.85 ± 0.75 m. On the other hand, SRL-KNN starts from the comparable result as LSTM model 5 with 0.90 m in the case of 0.5 m/s. After the maximum speed increases to above 1.5 m/s, the accumulated errors appear and the accuracy of SRL-KNN is significantly degraded to be above 3 m with the large variation being above 5 m.

I. UJIndoor Database

The consistent effectiveness of the proposed LSTM system is proved by the published dataset, UJIIndoorLoc [42]. The reported average localization error in [42] is 7.9 m. The database from 2 random phone users (Phone Id: 13, 14) in 2 different buildings (Building ID: 0 and 1) are used to implement LSTM. Note that the grid size of UJIIndoorLoc is different from the collected database and so is the average localization error. However, the relative accuracy comparison between the proposed LSTM and conventional KNN, e.g., RADAR [12] or feed-forward neural network, e.g, MLNN [21] can verify the effectiveness of our algorithm. Table V
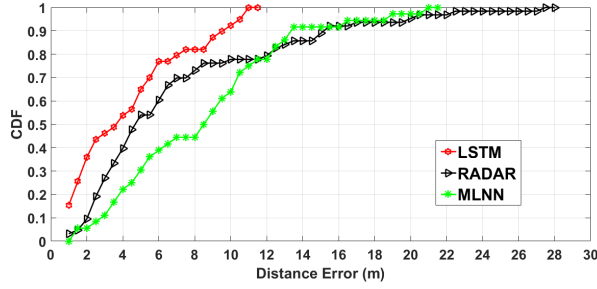
Fig. 9. Localization error CDF of UJIndoorLoc database for all buildings

shows the average errors in meter of LSTM model 5, RADAR, MLNN for each separate building and for all 2 buildings in general. For all 2 buildings, the average error of LSTM is 4.2 m ± 3.2 m, significantly lower than the result of RADAR 8.1 m ± 4.9 m and MLNN 7.5 m ± 3.8 m. Furthermore, Fig. 9 compares the CDF of localization errors between those 3 methods. In total, a 11.5 m maximum localization error is recorded for LSTM, 22 m for MLNN and the largest maximum localization error of 28 m for RADAR. Besides, 80% of the error is below 7 m in the case of LSTM, which is much lower than 12 m in the case of MLNN and RADAR.

V. CONCLUSIONS

In conclusion, we have proposed recurrent neural networks for fingerprint indoor localization using WiFi. Our RNN solution takes into account the relation between a series of the RSSI measurements and determines the user's moving path as one problem. Experimental results have consistently demonstrated that our LSTM structure achieves an average localization error of 0.75 m with 80% of the errors under 1 m, which outperforms feed-forward neural network, conventional methods such as KNN, Kalman filter and probabilistic methods. Furthermore, main challenges of those conventional methods including the spatial ambiguity, RSSI instability and the RSSI short collecting time have been effectively mitigated. Besides, the analysis of vanilla RNN, LSTM, GRU and BiLSTM with important parameters have been discussed in details, i.e, loss function, memory length, input and output features, etc.

APPENDIX

Let $D = d^2 = (x_i - x_{pre})^2 + (y_i - y_{pre})^2$. From (3):

$$\int_0^{d_{max}^2} P(l_i) = \int_0^{d_{max}^2} \frac{1}{2\sigma^2(1 - e^{\frac{d_{max}^2}{2\sigma^2}})} \exp(-\frac{D}{2\sigma^2}) d(D)$$

$$= \frac{1}{2\sigma^2(1 - e^{\frac{d_{max}^2}{2\sigma^2}})} (-2\sigma^2) \exp(-\frac{D}{2\sigma^2})\Big|_0^{d_{max}^2} = 1$$

REFERENCES

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 37, pp. 1067 – 1080, 2007.

[2] Y. Chapre, P. Mohapatra, S. Jha, and A. Seneviratne, "Received signal strength indicator and its analysis in a typical WLAN system (short paper)," *38th Annual IEEE Conference on Local Computer Networks*, 2013.

[3] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "Confi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18 066 – 18 074, Sep. 2017.

[4] X. Wang, X. Wang, and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 ghz Wi-Fi," *IEEE International Conference on Communications (ICC)*, pp. 1938–1883, Jul. 2017.

[5] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-modal deep learning for indoor localization with commodity 5ghz WiFi," *IEEE Access*, vol. 5, pp. 4209 – 4220, Mar. 2017.

[6] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. 3rd international conference on Mobile systems, applications, and services,Seattle, WA*, 2005, pp. 205–218.

[7] L. Chen, B. Li, and Z. Zheng, "An improved algorithm to generate a Wi-Fi fingerprint database for indoor positioning," *Sensors*, vol. 13, pp. 11 085–11 096, 2013.

[8] Y. Xu, *Autonomous Indoor Localization Using Unsupervised Wi-Fi Fingerprinting*. Kassel University Press, 2016.

[9] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Trans. Mobile Comput.*, vol. 6, pp. 689–705, June 2007.

[10] 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. 1250–1264, 2009.

[11] S. He and S.-H. G. Chan, "Wi-Fi fingerprint-based indoor positioning:recent advances and comparisons," *IEEE Communications Surveys & Tutorials*, vol. 18, 2016.

[12] P. Bahl and V. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," *IEEE Infocom*, 2000.

[13] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, "An improved K-Nearest-Neighbor indoor localization method based on Spearman distance," *IEEE Signal Processing Letters*, vol. 23, pp. 351 – 355, 2016.

[14] D. Li, B. Zhang, and ChengLi, "A feature-scaling-based K-Nearest Neighbor algorithm for indoor positioning systems," *IEEE Internet of Things Journal*, vol. 3, pp. 590 – 597, 2016.

[15] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Comp. Netw. ISDN Syst.47(6)*, vol. 47, pp. 825–845, 2005.

[16] K. Shi, Z. Ma, R. Zhang, W. Hu, , and H. Chen, "Support vector regression based indoor location in IEEE 802.11 environments," *Mobile Information Systems*, 2015.

[17] C.Gentile, N. Alsindi, R.Raulefs, and C.Teolis., *Geolocation Techniques Principles and Applications*. Springer, 2013.

[18] S.-H. Fang and T.-N. Lin, "Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments," *IEEE Transactions on Neural Networks*, pp. 1973–1978, Nov 2008.

[19] R. Battiti, A. Villani, and T. L. Nhat, "Neural network models for intelligent networks: deriving the location from signal patterns," *in: Proceedings of AINS2002, UCLA*, 2002.

[20] X. Lu, H. Zou, H. Zhou, L. Xie, and G.-B. Huang, "Robust extreme learning machine with its application to indoor positioning," *IEEE Transaction on Cybernetics*, vol. 46, no. 1, Jan. 2016.

[21] H. Dai, W. hao Ying, and J. Xu, "Multi-layer neural network for received signal strength-based indoor localization," *IET Communications*, vol. 10, Jan. 2016.

[22] 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.

[23] M. T. Hoang, Y. Zhu, B. Yuen, T. Reese, X. Dong, T. Lu, R. Westendorp, and M. Xie, "A soft range limited k-nearest neighbours algorithm for indoor localization enhancement," *IEEE Sensors Journal*, vol. 18, pp. 10 208–10 216, Dec. 2018.

[24] A. W. S. Au, C. Feng, S. Valaee, S. Reyes, S. Sorour, S. N. Markowitz, D. Gold, K. Gordon, and M. Eizenman, "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 2050–2062, 2013.

[25] I. Guvenc, C. Abdallah, R. Jordan, and O. Dedeoglu, "Enhancements to RSS based indoor tracking systems using Kalman filters," *Proc. Global Signal Processing Expo and Intl Signal Processing Conf*, 2003.

[26] A. Besada, A. Bernardos, P. Tarrio, and J. Casar, "Analysis of tracking methods for wireless indoor localization," *Proc. Second Intl Symp. Wireless Pervasive Computing (ISWPC 07)*, pp. 493–497, Feb 2007.

[27] A. Kushki, K. Plataniotis, and A. Venetsanopoulos, "Location tracking in wireless local area networks with adaptive radio maps," *Proc. IEEE Intl Conf. Acoustics, Speech and Signal Processing (ICASSP 06)*, vol. 5, pp. 741–744, May 2006.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[29] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoderdecoder approaches," *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, p. 103111, Oct. 2014.

[30] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673 – 2681, Nov. 1997.

[31] H. J. Jang, J. M. Shin, and L. Choi, "Geomagnetic field based indoor localization using recurrent neural networks," *GLOBECOM 2017*, 2017.

[32] R. Battiti, T. Nhat, and A. Villani, "Location-aware computing: a neural network model for determining location in wireless LANs," University of Trento , Trento, Tech. Rep., 2002.

[33] G. Huang, S. Song, C. Wu, and K. You, "Robust support vector regression for uncertain input and output data," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 23, pp. 1690–1700, Nov. 2012.

[34] Y. Lukito and A. R. Chrismanto, "Recurrent neural networks model for WiFi -based indoor positioning system," *International Conference on Smart Cities, Automation & Intelligent Computing Systems, Indonesia*, Nov. 2017.

[35] X. Wang, Z. Yu, and S. Mao, "DeepML: Deep LSTM for indoor localization with smartphone magnetic and light sensors," *IEEE International Conference on Communications (ICC)*, 2018.

[36] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv:1506.00019*, 2015.

[37] Y. Chapre, P. Mohapatray, S. Jha, and A. Seneviratne, "Received signal strength indicator and its analysis in a typical WLAN system," *IEEE Conference on Local Computer Networks*, 2013.

[38] K. Kaemarungsi and P. Krishnamurthy, "Analysis of WLANs received signal strength indication for indoor location fingerprinting," *Pervasive Mobile Computing*, vol. 8, pp. 292–316, 2012.

[39] P. Jiang, Y. Zhang, W. Fu, H. Liu, and X. Su, "Indoor mobile localization based on wi-fi fingerprint's important access point," *International Journal of Distributed Sensor Networks*, Apr. 2015.

[40] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of Applied Physiology*, vol. 100, pp. 390–398, Feb. 2006.

[41] B. J. M. Email and etc., "Visual flow influences gait transition speed and preferred walking speed," *Experimental Brain Research*, vol. 181, pp. 221–228, Aug. 2007.

[42] J. Torres-Sospedra and etc., "Ujiindoorloc: A new multi-building and multi floor database for wlan fingerprintbased indoor localization problem," *Indoor Positioning and Indoor Navigation (IPIN)*, 2014.