Towards the Implementation of Recurrent Neural Network Schemes for WiFi Fingerprint-Based Indoor Positioning

He-Yen Hsieh, Setya Widyawan Prakosa, Jenq-Shiou Leu, *Senior Member, IEEE*
Department of Electronic and Computer Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
{m10502103, m10502804, jsleu}@mail.ntust.edu.tw

***Abstract*—The rapid development of Indoor Positioning System has attracted researcher to develop a robust scheme to predict the location based on Received Signal Strength Indicator (RSSI) signal. A lot of research topics presented in many journals and conferences by many researchers concern indoor positioning system as a main topic [1], [2]. Currently, the study related to find the robust algorithm for indoor positioning system becomes a high demand topic in several conferences. Our work intents to evaluate the effectiveness of Recurrent Neural Network (RNN) as a deep learning technique to be implemented in this field. In addition, LSTM as a variant of RNN scheme is also implemented. The purpose of this implementation is to explore both LSTM and original RNN to be utilized for localization in indoor positioning scheme, especially for Wifi Fingerprinting Dataset. From all evaluations, our proposed approach could get 99.7% accuracy for predicting which floor the sensor belongs to. In addition, the distance errors of our scheme are around 2.5 - 2.7 meters.**

***Index Terms*—Indoor Positioning System, Received Signal Strength Indicator (RSSI), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM).**

I. Introduction

As the rapid advancement of wireless technology, Wireless Local Area Networks (WLANs), commonly called Wi-Fi, are broadly used in urban scenarios, particularly for Indoor Positioning. WiFi is a popular technology to realize a broadband communications, especially for human communications and machine-to-machine schemes. The ease to employ WLANs as broadband communication technology make this technology more popular. Thus, Wireless Sensor Network (WSN) becomes widely used on the application of the technology and many research topics related to classification of WSN have been conducted and the difficulty of classification task for WSN has been assessed on several works. K-Means Clustering as the classifier has been well implemented to classify Received Signal Strength (RSS). Basic and Hybrid-schemes to employ K-Nearest Neighbor (KNN) for WLAN based Indoor Positioning System have been proposed by Sun *et-al* [3]. The approach is to combine KNN and Fuzzy Clustering Means (FCM). FCM is implemented to cluster determined by K-nearest neighbours then the calculation of user position will be counted from the selected one of the clusters.

As the introduction of wireless sensor for Indoor Positioning, the development of algorithm for classifying WiFi signal becomes essential to be developed. The current survey [4] has confirmed that the common result of tested methods for distance error of WiFi-based fingerprinting approaches is around 2-5 meters. In addition, RNN is successfully implemented in various application and the results from some reports indicate that RNN is a powerful deep learning scheme [5], [6]. Almost all of the implementations reach high accuracy, it means that RNN is a promising deep learning technique to be implemented in various areas. In this study we try to implement Deep Learning algorithm which is RNN to reach the high accuracy results for classification scheme.

The utilization of other variant RNN, namely Long Short Term Memory (LSTM), gives promising results in various applications. LSTM has been successfully applied for detecting the trajectory of Indoor Pedestrian and image based localization [7], [8]. From both reports, it can be clearly seen that, LSTM gives a promising outcome for both tasks. Furthermore, we are inspired by these works to apply LSTM in indoor positioning system. To the best of authors' knowledge, our work is the first for handling the issue related to the prediction of indoor positioning system, particularly for WiFi fingerprint-based indoor positioning, using original RNN and LSTM. This preliminary study will emerge a new idea for exploring the construction of robust scheme in this field utilizing deep learning technique.

In this paper, a design of classifier scheme for RSS-based indoor positioning using RNN is presented. We also show the implementation of LSTM as the variant of RNN. Evaluation between both methods is also presented to give a clear report related to the performance of RNN for indoor positioning scheme. From the experiment results, both of designs implemented original RNN and LSTM can reach high accuracy. Furthermore, the remainder of this paper is organized as follows. Some backgrounds related to this topic are surveyed in section 2. Section 3 reviews the design and the detail of our proposed scheme. While section 4 explains the performance of the proposed schemes and section 5 discusses the future work and concludes the proposed work.

II. Background

A. Indoor Positioning Dataset

To verify our proposed scheme, validation and test for the performance of it is conducted on dataset released online. The authors have published the dataset as a Long-Term Wifi Fingerprinting Dataset [9]. As stated on the report, the purpose of the dataset is to ease researchers who are interested to devote their research on WiFi fingerprinting based indoor positioning system.

WiFi Fingerprinting based indoor localization dataset comprises of 46800 data for testing and 16704 for training. Testing data and training data are different. According to the paper, the dataset has been standardized. Furthermore, the dataset has also been a benchmark for indoor localization problem.

The WiFi for indoor localization has been deployed on two floors and each sensor is transmitting the x and y data (cartesian coordinate) that tells us the exact location based on RSSI. In this paper, we will show the effectiveness of our approach to achieve low error prediction for x and y position of each sensor as well as to build an accurate model to determine which floor the sensor belongs to.

B. Recurrent Neural Network (RNN)

Currently, Recurrent Neural Networks (RNN) has been employed for classification algorithm on several applications including medical diagnosis, stock exchange, and natural language processing. RNN is powerful to handle time series data, however there are no researches focusing on employing RNN for Indoor Positioning scheme.

To deal with the aforementioned database, we design the implementation to classify x and y coordinate based on RSSI of each sensor. Furthermore our scheme is also able to recognize which floor of each sensor belongs to. For detecting the cartesian coordinate of each sensor, we employ RNN regression. As the usabilty of regression scheme which is feasible for predicting the value of x and y of each node, we regard the position of the wireless sensor as a node. In addition, the classification scheme used to predict the floor is RNN classification.

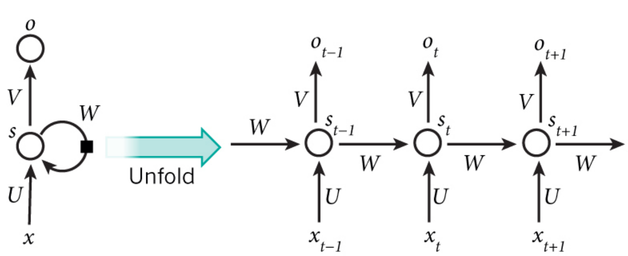


Fig. 1: Recurrent Neural Network (RNN) basic structure [10]

From picture 1, the unfolded RNN comprises of n-layer which can be constructed to build a deep network. Here, x_t is the input for every time step t where each time t has the hidden state which is the "memory" of the network called s_t. The hidden state s_t can be counted based on the prior hidden state and the input at the current step. o_t is an output from step t. Mathematically, we can show the "memory" counted as,

$$s_t = f(Ux_t + Ws_{t-1}) \tag{1}$$

Normally, deeper structure will consume more times to train and test it. In addition, a deeper structure is easily trapped on overfitting due to a few number of training data then we would have a low error on training but high error when we test it. The study related to how effectiveness of the depth for the learning scheme will be shown in the following section which is going to discuss and evaluate if we apply more RNN layers.

C. Long Short Term Memory Networks

As the variant of RNN, LSTM is applicable to handle time series data. Commonly, LSTM consists of a memory cell, a forget gate, an input and output gate. It also has an advantage of the forget gate and memory cell in the structure of LSTM because we can easily control when the data should be buffered or released. The main concept of LSTM is to handle the RNN flaw's to tackle the issue related to avoiding the long-term dependency problem.

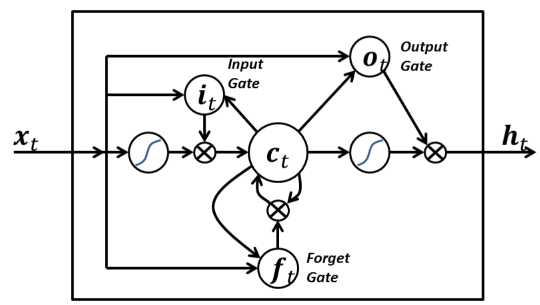


Fig. 2: Basic structure for Long Short Term Memory (LSTM) [11]

Figure 2 depicts a structure of LSTM that consists of f_t as a forget gate. The memory cell is indicated by c_t where an input gate is i_t and o_t as an output gate. Below equations are the expression for mathematical modelling of LSTM network.

$$f_t = \sigma_g(W_f x_t + U_f c_{t-1} + b_f) \tag{2}$$

$$i_t = \sigma_g(W_i x_t + U_i c_{t-1} + b_i) \tag{3}$$

$$o_t = \sigma_g(W_o x_t + U_o c_{t-1} + b_o) \tag{4}$$

In the evaluation for practical implementation of LSTM, we try several parameter for bias on forget gate. Parameter adjustment might affect the accuracy of the trained model. In this case our purpose is to clarify whether the parameter can influence the final model for Indoor Positioning System or it just has a less contribution for final model. Section IV (results and discussion of the proposed scheme performance) will review some parameters that we have tried and summarize the outcomes.

III. ALGORITHM DESIGN

A. Detail Implementation of Proposed Scheme

The entire design of localization indoor positioning system using RNN and LSTM is described in picture 3. Firstly, the database obtained from public domain is stored in our server. Since the dataset is already well arranged by authors then we can directly put it in our scheme. There are two fully connected layers, one has 40 hidden layers and the other has 2. Before processing into fully connected layers, firstly, the dataset is stored into RNN or LSTM regression. The layers of LSTM and RNN are adjusted and we use 1, 3, and 5 layers. Some adjustment is essential to analyze the effect of the layer numbers whether it can boost the performance or worsen the accuracy. The regression model is used to predict x and y value of each sensor. Thus, the classification model predicts the floor.

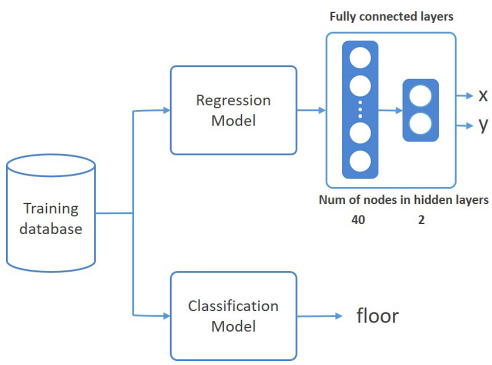


Fig. 3: Detail Implementation of the Proposed Scheme

B. RNN Training and Testing

Figure 4 depicts how we begin the training stage and generate the model. Training begins with the preparation for database. From the database we train the data based on regression to get the model for predicting x and y value. To classify the floor, we employ LSTM or RNN classification method. Furthermore, after training is finished, the model is stored in our server and finally the model is tested to verify the accuracy obtained by the scheme entirely.

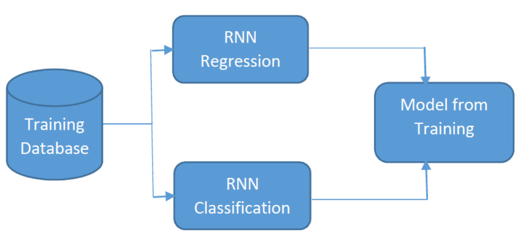


Fig. 4: Training Phase

After we generate the model, verification to get the accuracy of the entire scheme is conducted using the scheme depicted by figure 5. To verify, the model generated from training stage is tested using testing dataset that is different from training data. The model is expected to predict the right value from testing dataset to get coordinates of each sensor as well as predict which floor each sensor belongs to. After the model predicts the coordinates and floor, using the ground truth we can calculate the accuracy of the model. Furthermore, we can discover whether the proposed scheme is accurate enough or not.

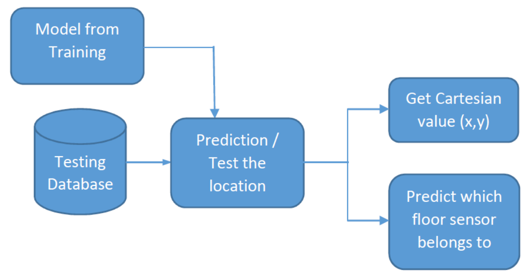


Fig. 5: Testing Phase

IV. RESULTS AND DISCUSSION OF THE PROPOSED SCHEME PERFORMANCE

Tensorflow [12] is implemented in python 3.5 to conduct all experiments in this proposed work. On the other hand, we use server with Ubuntu 16.04 as operating sytem (OS) and NVIDIA GeForce GTX 1080 Ti as Graphical Processing Unit (GPU).

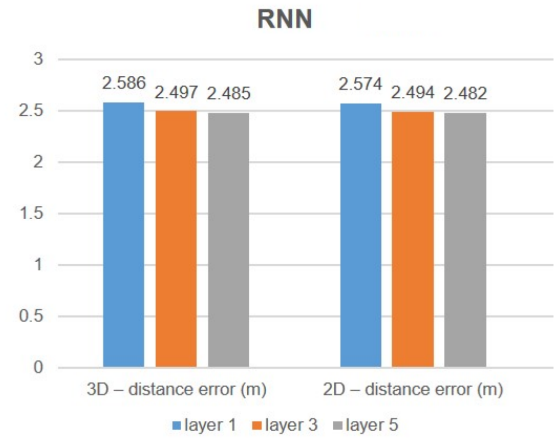
A. Experiment on RNN

Outcomes from the proposed scheme using basic RNN have three indicators shown in the chart such as 3D-distance error, 2D-distance error, and floor accuracy. 3D-distance error is calculated without considering error from floor classification. In addition, if we calculate distance error with the consideration of floor detection, the distance error is shown in 2D-distance error. Normally, 3D-distance error is higher than 2D-distance error, because in the calculation of 2D distance error, if we find error in the floor classification in a specific point then we do not count this point in the calculation of 2D distance error. Picture 6 depicts the outcomes from proposed scheme using RNN. The performance of deeper layer is more precise to predict floor and coordinates.

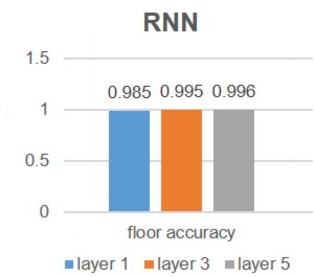
B. Experiment on LSTM

Figure 7 depicts the experiment results using 1 layer LSTM. From the figure, we implement two values for forget gate bias (0.8 and 1). We also employ bidirectional LSTM to clarify either using different LSTM can improve the performance or degrade it. The outcomes shown in figure 7 indicates that, there are a slightly difference between the outcomes of LSTM with forget bias 0.8 and 1. Even if we employ bidirectional LSTM, the difference is negligible which is around 0.027 meters for distance error and 0.001 for floor classification.

Applying more layers for LSTM to gain more accurate model is a common approach in the deep learning technique. Figure 8 and 9 show the results for 3 layer and 5 layer

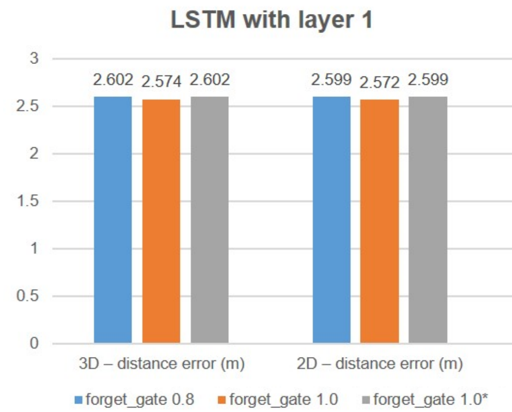


(a) Distance error

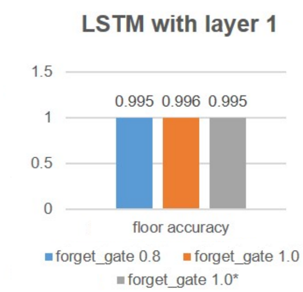


(b) Floor detection accuracy

Fig. 6: Distance error and floor detection accuracy using RNN



(a) Distance error



(b) Floor detection accuracy

Fig. 7: Distance error to predict coordinates and floor detection accuracy for LSTM using 1 layer

LSTM schemes. However, the accuracy gained from both schemes is not too far from LSTM with 1 layer. There is little improvement obtained by employing more layers.

C. Computation Time

The computational costs for RNN and LSTM with different layer settings are shown in table I. More added layers for both RNN and LSTM will cost longer training and testing time. As presented in table I, 1 layer RNN needs 564.1396s to train and creates a model and 10.0848s for testing time. Compared to 1 layer RNN, 3 layer RNN requires about 300s and 2s longer for training and testing.

TABLE I: Measurements of Computation Time

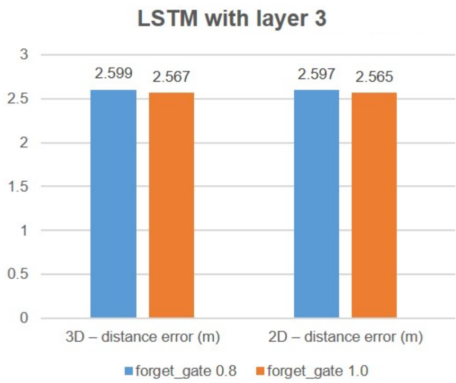
Model	Layer	Training Time (s)	Testing Time (s)
RNN	1	564.1396	10.0848
RNN	3	726.9329	12.0119
RNN	5	869.6197	12.0286
LSTM	1	581.3599	10.1721
LSTM	3	782.9536	12.0249
LSTM	5	946.3452	13.5178

D. Discussion

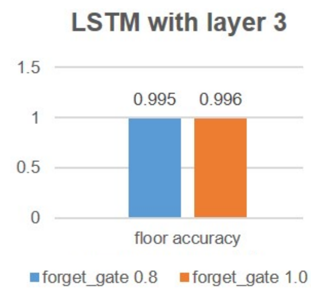
Using RNN and LSTM, we obtain an accurate prediction to detect which floor the sensor belongs to and obtain distance error around 2.5 - 2.7 meters. Outcomes from RNN indicate that by adding layers, from 1 to 5 layers, can increase the accuracy. Using 1-layer RNN, the accuracy to predict floor is 98.5% and it can be improved by adding the layer to 3 or 5 layers to reach 99.5% for 3-layer and 99.6% for 5-layer. However, by adding some layers to the proposed scheme will increase the computational process as presented in table I. For LSTM, the performance of 1, 3, and 5-layer LSTM is not as far as RNN if we employ more layers which is just 0.027 meters for distance error and 0.001% for floor prediction, even if we employ different forget bias. Furthermore, our proposed scheme using RNN and LSTM as one of deep learning techniques gives a promising result. We could obtain 99.7% accuracy to predict floor using 5-layer LSTM and all evaluations for our implementation indicate that the distance errors for the entire approaches are around 2.5 - 2.7 meters.

V. Conclusion and Future Work

From all of discussions in this paper, we could show that using deep learning technique, here we use RNN and LSTM, we could obtain promising results for indoor positioning system based on WiFi fingerprinting dataset. The distance errors for all evaluations are around 2.5 -2.7 meters and we also reach 99.7% as the best result accuracy gained by employing 5-layer LSTM. Even though the proposed approach

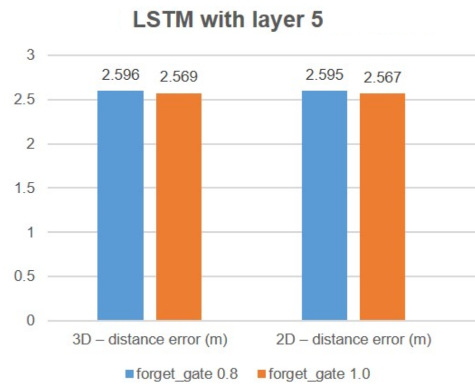


(a) Distance error

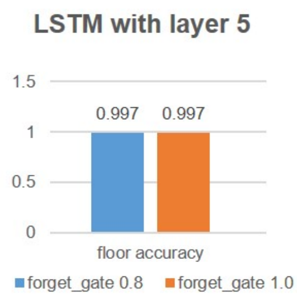


(b) Floor detection accuracy

Fig. 8: Distance error to predict coordinates and floor detection accuracy for LSTM using 3 layers



(a) Distance error



(b) Floor detection accuracy

Fig. 9: Distance error to predict coordinates and floor detection accuracy for LSTM using 5 layers

can obtain promising results, the improvement to enhance the performance of our approach is essential, particularly to create a robust model that has a lower distance error. The utilization of statistical method especially for preprocessing the dataset might be useful to enhance the proposed approach performance.

Acknowledgement

The authors gratefully acknowledge the support by the Ministry of Science and Technology, Taiwan, under grant MOST-106-3114-E-011-009

References

[1] X. Wang, L. Gao, S. Mao, S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," IEEE Transactions on Vehicular Technology, vol. 66, pp. 763–776, 2017.

[2] Y. Tian, D. Shigaki, W. Wang, C-J Ahn, "A weighted least-squares method using received signal strength measurements for WLAN indoor positioning system," 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 310–314, 2017.

[3] Y. Sun, Y. Xu, L. Ma, Z. Deng, "KNN-FCM hybrid algorithm for indoor location in WLAN," 2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS), vol. 2, pp. 251–254, 2009.

[4] S. He, S. H. G. Chan, "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons," IEEE Communications Surveys & Tutorials, vol. 8, pp. 466–490, 2016.

[5] H. J. Jang, J. M. Shin, L. Choi, "Geomagnetic Field Based Indoor Localization Using Recurrent Neural Networks," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp. 1–6, 2017.

[6] X. Gan, B. Yu, L. Huang, Y. Li, "Deep learning for weights training and indoor positioning using multi-sensor fingerprint," 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–7, 2017.

[7] F. Walch, C. Hazirbas, L. Leal-Taix, T. Sattler, S. Hilsenbeck, D. Cremers, "Image-Based Localization Using LSTMs for Structured Feature Correlation," 2017 IEEE International Conference on Computer Vision (ICCV), pp. 627–637, 2017.

[8] J. Li, Q. Li, N. Chen, Y. Wang, "Indoor Pedestrian Trajectory Detection with LSTM Network," 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 651–654, 2017.

[9] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta, "Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning," Data, January 2018.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, 521, pp 436-444, 10.1038/nature14539, 2015.

[11] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM Recurrent Networks," J. of Machine Learning Research, vol. 3, pp. 115143, 2002.

[12] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Man, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Vigas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," Software available from tensorflow.org, 2015.