2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)

# Indoor Pedestrian Trajectory Detection with LSTM Network

Jin Li
University of Chinese Academy of Sciences
ShanghaiTech University
Shanghai, China
lijin@shanghaitech.edu.cn

Qiang Li, Nanxi Chen,Yingguan Wang
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
Shanghai, China
qiang.li@mail.sim.ac.cn
nanxi.chen@wico.sh
wyg@mail.sim.ac.cn

***Abstract*— This paper proposes a novel technique to detect the main moving trajectory of indoor pedestrians. Based on Long Short-Term Memory(LSTM) Network, this deep learning network is capable of learning the trajectory of human beings using indoor Wi-Fi positioning data. The data is collected by Wi-Fi detectors densely installed in a public building in the urban area, which can ensure the detection of any portable devices as long as the Wi-Fi module is turned on. Then the model works in the form of sequence modeling to learn the trajectory of the main stream extracted from massive pedestrian positioning data. In compare with methods like Recurrent Neural Network (RNN) and Gated Recurrent Unit(GRU), there is an obvious performance improvement of this method.**

***Keywords*—Deep Learning;Long Short Term Memory; Wi-Fi Position; Recurrent Nuearal Network**

## I. Introduction

LSTM (Long Short-Term Memory) presented by Hochreiter & Schmidhuber[1]is a special kind of RNN (Recurrent Neural Networks). It preserves the advantages of the RNN which take advantage of directional circulation that enable it to deal with the problem of context relationship between inputs. In addition, it also solves the Long-Term Dependencies problem that RNN can only use a short steps of previous information of a sequence data. As a result, LSTM are useful at capturing long-term temporal dependences without suffering the optimization problems of RNN. The most significant point of LSTM architecture is its memory cell which can maintain the state over time. In addition, there are non-linear gating units which can regulate the information flow into and out of the cell. In 2015, Klaus Greff et- al.[2] wrote a survey about basic LSTM and LSTM variants. It shows the detailed architecture of LSTM and reveals impact of the change of hyperparameters on the LSTM performance variance. Figure 1 is extracted from this survey. It exhibits the structure of LSTM and a simple structure of RNN. Let $x_t$ denote the input vector at time t, W denote the rectangular input weight matrices, R denote square recurrent weight matrices, p denote the peephole weight vectors and b denote the bias vectors. And functions $\sigma$, g and $h$ are the representations of point-wise non-linear activation function: logistic sigmoid ($\frac{1}{1+e^{-x}}$), which is used for the activation of gates. The hyperbolic tangent is used as the block input and output activation function. The point-wise multiplication of two vectors is represented by $\odot$:

$$z^t = g(W_z X^t + R_z y^{t-1} + b_z) \qquad \text{block input}$$

$$i^t = \sigma(W_i X^t + R_i y^{t-1} + p_i \odot c^{t-1} + b_i) \qquad \text{input gate}$$

$$f^t = \sigma(W_f X^t + R_f y^{t-1} + p_f \odot c^{t-1} + b_f) \qquad \text{forget gate}$$

$$c^t = i^t \odot z^t + f^t \odot c^{t-1} \qquad \text{cell state}$$

$$o^t = \sigma(W_o X^t + R_o y^{t-1} + p_o \odot c^{t-1} + b_o) \qquad \text{output gate}$$

$$y^t = o^t \odot h(c^t) \qquad \text{block output}$$

Another work is done by Andrej Karpathy et al.[3]. They compare the performance of LSTM, RNN and GRU proposed by Cho et al.[4] based on Penn Treebank dataset[5] and Hutter Prize 100MB of Wikipedia dataset[6].

RNNs including LSTM have a variety of applications. Graves et al.[7], Pham et al.[8] and Doetsch et al.[9] apply it to handwriting recognition. Graves et al.[10] also use it for handwriting generation. There are a lot of other applications like text generation (J. Martens et al.[11]), Music composition (I.-T. Liu et al.[12]), language modeling (Zaramba et al.[13]) and translation (Luong et al.[14]), acoustic modeling of speech (Sak et al.[15]), Speech synthesis ( Fan et al.[16]) and video data(Donahue et al.[17]).

This paper is inspired by Rajiv C. Shah et al.[18] and Alexandre Alahi et al.[19]. Rajiv c. Shah et.al apply deep learning to basketball trajectories. They use a 2 layers LSTM network of 64 hidden unites. Each input data is combined of 3D coordinate data and time point data as a 4D vector. The label of each data is whether the basketball shots or not. After training the network with Adam optimizer of a learning rate 0.005, dropout rate 0.6 and a batch sizer 64, it can predict if a basketball can shot or not. They address a classification problem with coordinate data.

Alexandre Alanhi et al. propose a new model called 'Social LSTM' to jointly predict the paths of all the people in a scene. What they address is a tracking problem with video data. They take into account the common sense rules and social conventions that humans typically utilize as their navigate in shared environment.

Based on those results, this paper focuses on detecting the main stream trajectory of indoor pedestrian.

978-1-5386-3221-5/17 $31.00 © 2017 IEEE
DOI 10.1109/CSE-EUC.2017.122

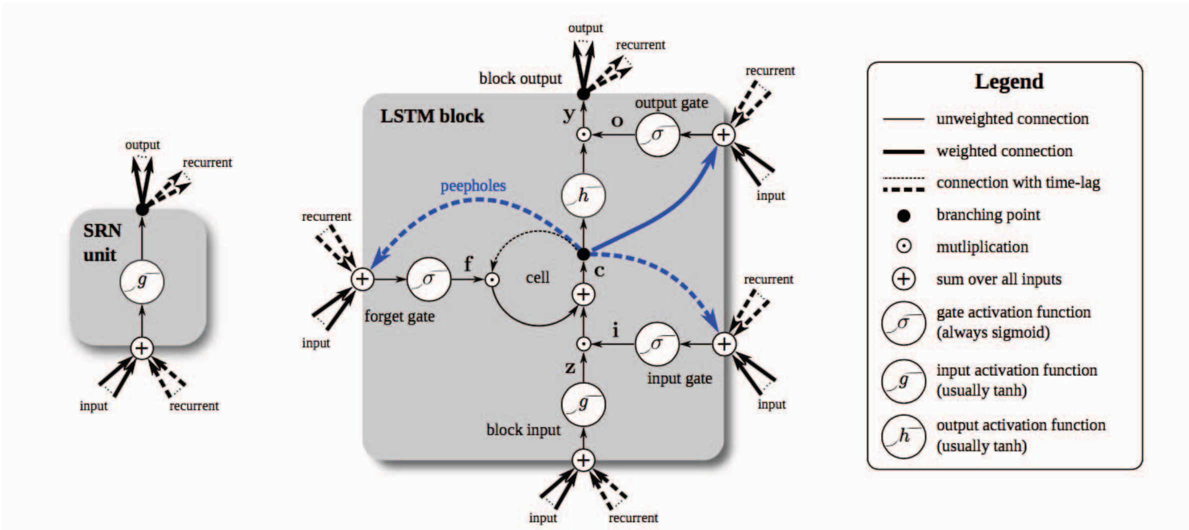


Figure 1 Detailed schematic of the Simple Recurrent Network (SRN) unit (left) and a Long Short-Term Memory block (right) as used in the hidden layers of a recurrent neural network.

Firstly, according to the work done by Dongkuan Xun and Yingjie Tian[20], this paper applies DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[21] to extract main stream coordinate of all coordinate data. The trajectory of a device can be recognized from its MAC Address, which can be used to distinguish a set of trajectories from each other.

Secondly, the trajectory is non-linear and it makes LSTM a better choice for our prediction, rather than a traditional linear model. Taking advantage of this feature, this paper will use LSTM to detect the trajectory.

## II. Approach

### A. Data Preparation

The data is collected with the device Wi-Fi sniffer we made. In a rectangle house the coordinate is establish and Wi-Fi sniffers are arranged like Figure 2. With the dense distribution of Wi-Fi detectors, this paper uses trilateration positional method to measure the relationship between RSS and distance. The equation connecting Received Signal Strength (RSS) and Distance is showed below:

$$RSS = -10n\log(d) - A \tag{1}$$

where n is the path loss factor, d is the distance . A is the reference RSS value that a portable device received at 1 meter away from the Wi-Fi sniffer.

Figure 3 shows the diagrammatic sketch.

With all the Wi-Fi sniffers arranged by ourselves, the Wi-Fi detector 1, 2 and 3 can has coordinate $(x_1, y_1)$, $(x_2, y_2)$ and $(x_3, y_3)$, respectively. The root of arrow is a device that to be detected. Its coordinate is $(x, y)$ and it has different distance $d_1$, $d_2$ and $d_3$ to Wi-Fi detector 1,2 and 3. Now equations are listed below:

$$(x - x_1)^2 + (y - y_1)^2 = {d_1}^2 \tag{2}$$

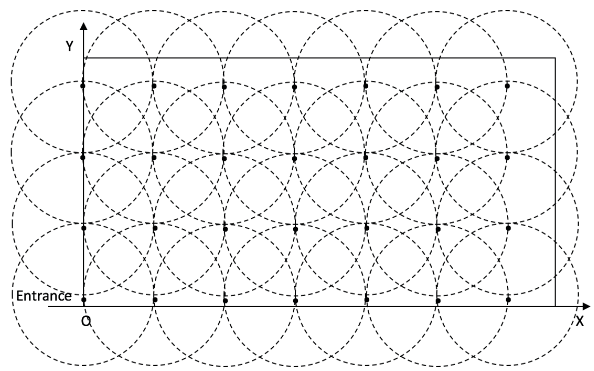


Figure 2 Diagrammatic sketch of the distribution of Wi-Fi detector

$$(x - x_2)^2 + (y - y_2)^2 = {d_2}^2 \tag{3}$$

$$(x - x_3)^2 + (y - y_3)^2 = {d_3}^2 \tag{4}$$

Using equation (2) minus equation (4) and equation (3) minus equation (4), they become equation (5) and equation (6):

$$2(x_1 - x_3)x + 2(y_1 - y_3)y = {x_1}^2 - {x_3}^2 + {y_1}^2 - {y_3}^2 + {d_3}^2 - {d_1}^2 \tag{5}$$

$$2(x_2 - x_3)x + 2(y_2 - y_3)y = {x_2}^2 - {x_3}^2 + {y_2}^2 - {y_3}^2 + {d_3}^2 - {d_2}^2 \tag{6}$$

let:

$${x_1}^2 - {x_3}^2 + {y_1}^2 - {y_3}^2 + {d_3}^2 - {d_1}^2 = r_1 \tag{7}$$

$${x_1}^2 - {x_3}^2 + {y_1}^2 - {y_3}^2 + {d_3}^2 - {d_1}^2 = r_2 \tag{8}$$

so:

$$[(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3)]y = (x_2 - x_3)r_1 - (x_1 - x_3)r_2 \tag{9}$$

let:

$$(x_2 - x_3)(y_1 - y_3) - (x_1 - x_3)(y_2 - y_3) = m \tag{10}$$

$$(x_2 - x_3)r_1 - (x_1 - x_3)r_2 = n \tag{11}$$

so：

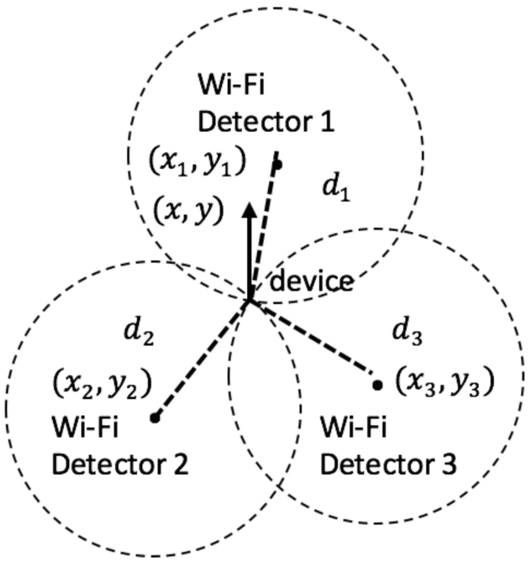


Figure 3:Diagrammatic sketch to count coordinates of electronic device

$$y = \frac{n}{m} \tag{12}$$

$$x = \frac{r_1 - \frac{n}{m}(y_1 - y_3)}{(x_1 - x_3)} \tag{13}$$

This is the coordinate of one point of the trajectory.

With the distribution of Wi-Fi sniffers, when human beings walk sniffers can detect a sequence of points. It is the coordinate representation of the trajectory that a human generates. Each trajectory has 50 points. x coordinate is used as the input data, y coordinate as the ground truth.

The amount of data collected is 200,000 and it is divide into two parts: 160,000 for training and 40,000 for testing.

### B. *Long Short Term Memory Neural Network*

To show the structure of our LSTM, this paper uses tool called TensorBoard integrated in TensorFlow [22] to exhibit it.

The general architecture diagram of this neural network is shown in Figure 4.

Inputs is the place where data is imported. In_hidden layer and out_hidden layer have the same transformation structure. In_hidden layer is the transformation of inputs, which is function (14):

$$\mathrm{y}_{\mathrm{in\_hidden}} = \mathrm{W}_{\mathrm{in\_hidden}} \mathrm{x}_{\mathrm{in}} + \mathrm{b}_{\mathrm{in\_hidden}} \tag{14}$$

$\mathrm{W}_{\mathrm{in\_hidden}}$ is the rectangular input weight matrices.

x is the input data.

$\mathrm{b}_{\mathrm{in\_hidden}}$ is bias vectors.

$\mathrm{y}_{\mathrm{in\_hidden}}$ is the output of the transformation.

As for out_hidden layer, it is

$$\mathrm{y}_{\mathrm{out\_hidden}} = \mathrm{W}_{\mathrm{out\_hidden}} \mathrm{x}_{\mathrm{LSTM}} + \mathrm{b}_{\mathrm{outhidden}} \tag{15}$$

where $\mathrm{x}_{\mathrm{LSTM}}$ is the output of LSTM_cell.

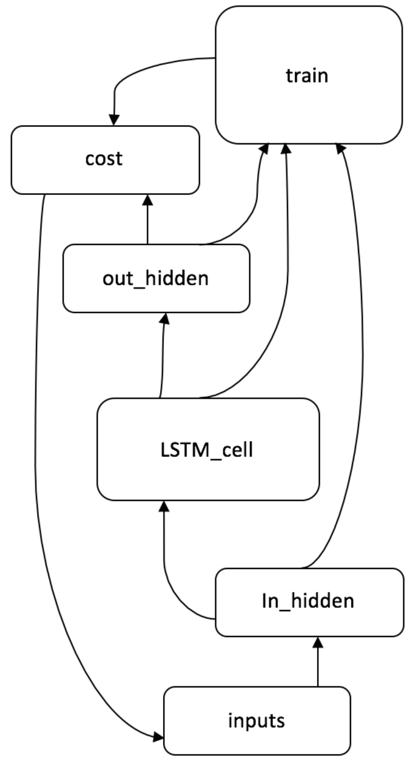


Figure 4: General architecture diagram of Neural Network

The LSTM block is the most important part of the architecture. It has two LSTM layers. Each layer has 50 basic LSTM units.

The cost function is Mean Square Error (MSE), which is a general cost function to measure the average error between result predicted and the ground truth.

Yoshua Bengio[23] gives a comprehensive practical recommendations for training deep architectures. According to this article, the network is trained with Adam algorithm [24] with a learning rate of 0.006, dropout rate of 0.5, and batch size of 50.

## III. EXPERIMENTAL RESULT

The goal is to detect the main stream of pedestrian trajectory in a hot spot place. To measure the advantage of our model, its result of accuracy is compared with two basic methods: basic RNN model and GRU.

The result is showed in Figure 5. Horizontal axis shows that through the whole testing process, it outputs 10 points. While the vertical axis exhibits the cost value of those points.

Figure 5 shows three lines. The solid line represents the cost of GRU Neural Network. The dotted line represents the cost of our LSTM Neural Network. At last the dotted line with triangle represents basic RNN Neural Network. As the process proceeds, costs come into a stationary phase. Actually, the final cost value of GRU is 2.5656 and RNN's cost value is 2.4661. In compare with those two values, the cost value of LSTM Network presented in this paper is 1.8603, which is much lower than those

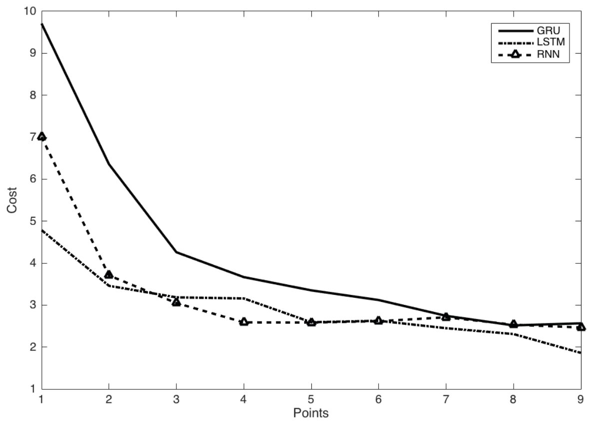


Figure 5: Experiment result

two. It has an improvement of 27% compared with GRU Neural Network and an improvement of 28% compared with basic RNN Neural Network. This means that our LSTM Neural Network is more accurate than those two Neural Networks in this main stream pedestrian trajectory detection.

## IV. CONCLUSION

This paper develops a two layers LSTM Neural Network models for the detection of main stream pedestrian trajectory with Wi-Fi positioning data. In contrast with normal Neural Networks like RNN and GUR, it has an obvious improvement. The cost of our Neural Network is 1.8603 while GRU's is 2.5658 and RNN's is 2.4661,which is an obvious improvement .

This paper applies LSTM to motion tracking data. The results here suggest LSTM has the ability to offer a utilization of trajectory sequence data.

Future work will likely to combine it with other technology like tracking to offer a more accurate trajectory result.

## REFERENCES


[1] Hochreiter, Sepp and Schmidhuber, Ju ̈rgen. Long short-term memory. Neural computation, 9(8): 1735–1780, 1997.

[2] Greff, Klaus, Srivastava, Rupesh Kumar, Koutn ́ık, Jan, Steunebrink, Bas R., and Schmidhuber, Ju ̈rgen. LSTM: A Search Space Odyssey. CoRR, abs/1503.04069, 2015. URL http://arxiv. org/abs/1503.04069.

[3] Andrej Karpathy, Justin Johnson, and Li Fei Fei. Visualizing and Understanding Recurrent Networks. CoRR. abs/1506.02078v2,2015.URLhttps://arxiv.org/pdf/1506.02078v2.pdf

[4] Cho, Kyunghyun, van Merrie ̈nboer, Bart, Bahdanau, Dzmitry, and Bengio, Yoshua. On the proper- ties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259, 2014.

[5] Marcus, Mitchell P, Marcinkiewicz, Mary Ann, and Santorini, Beatrice. Building a large annotated corpus of english: The penn treebank. Computational linguistics, 19(2):313–330, 1993.

[6] Hutter, Marcus. The human knowledge compression contest. 2012.

[7] Graves, A, Liwicki, M, Fernandez, S, Bertolami, R, Bunke, H, and Schmidhuber, J. A Novel Connectionist System for Improved Unconstrained Handwriting Recognition. IEEE Transactions on Pattern Analysis and Machine In- telligence, 31(5), 2009.

[8] Pham, Vu, Bluche, The ́odore, Kermorvant, Christopher, and Louradour, Je ́roˆme. Dropout improves Recur- rent Neural Networks for Handwriting Recognition. arXiv:1312.4569 [cs], November 2013. URL http: //arxiv.org/abs/1312.4569.

[9] Doetsch, Patrick, Kozielski, Michal, and Ney, Hermann. Fast and robust training of recurrent neural networks for offline handwriting recognition. In 14th International Conference on Frontiers in Handwriting Recognition, 2014. URL http://people.sabanciuniv. edu/berrin/cs581/Papers/icfhr2014/ data/4334a279.pdf.

[10] Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Ge- offrey. Speech recognition with deep recurrent neural networks. arXiv preprint arXiv:1303.5778, 2013. URL http://arxiv.org/abs/1303.5778.

[11] J. Martens. Generating Text with Recurrent Neural Networks. Neural Networks, 131(1):1017–1024, 2011.

[12] I.-T. Liu and B. Ramakrishnan. Bach in 2014: Music Composition with Recurrent Neural Network. arXiv:1412.3191, 5:1–9, 2014

[13] Zaremba, Wojciech, Sutskever, Ilya, and Vinyals, Oriol. Recurrent Neural Network Regularization. arXiv:1409.2329 [cs], September 2014. URL http: //arxiv.org/abs/1409.2329.

[14] Luong, Thang, Sutskever, Ilya, Le, Quoc V., Vinyals, Oriol, and Zaremba, Wojciech. Addressing the Rare Word Problem in Neural Machine Translation. arXiv preprint arXiv:1410.8206, 2014. URL http://arxiv.org/ abs/1410.8206.

[15] Sak, Hasim, Senior, Andrew, and Beaufays, Franc ̧oise. Long short-term memory recurrent neural network archi- tectures for large scale acoustic modeling. In Proceed- ings of the Annual Conference of the International Speech Communication Association (INTERSPEECH), 2014. URL http://193.6.4.39 ̃czap/letoltes/IS14/IS2014/PDF/AUTHOR/IS141304.PDF.

[16] Fan, Yuchen, Qian, Yao, Xie, Fenglong, and Soong, Frank K. TTS synthesis with bidirectional LSTM based recurrent neural networks. In Proc. Interspeech, 2014.

[17] Doetsch, Patrick, Kozielski, Michal, and Ney, Hermann. Fast and robust training of recurrent neural networks for offline handwriting recognition. In 14th International Conference on Frontiers in Handwriting Recognition, 2014. URL http://people.sabanciuniv. edu/berrin/cs581/Papers/icfhr2014/ data/4334a279.pdf.

[18] Rajiv C. Shah and Rob Romijnders. Applying Deep Learning to Basketball Trajectories. arXiv preprint arXiv:1608.03793v2,2016. URL https://arxiv.org/pdf/1608.03793.pdf

[19] Alexandre Alahi∗ , Kratarth Goel∗ , Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, Silvio Savarese. Social LSTM: Human Trajectory Prediction in Crowded Spaces. arXiv preprint arXiv:1611.09464v1,2016. URL https://arxiv.org/pdf/1611.09464v1.pdf

[20] Dongkuan Xu and Yingjie Tian. A Comprehensive Survey of Clustering Algorithms. Annals of Data Science.2015

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu .A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.

[22] Martín Abadi, Ashish Agarwal, Paul Barham,et al. Tensorflow: Large-Scale Machine Learn on Heterogeneous Distributed System. arXiv preprint arXiv:1603.04467, 2016. URL https://arxiv.org/pdf/1603.04467v2.pdf

[23] Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures. . arXiv preprint arXiv:1206.5533v2, 2012. URL https://arxiv.org/pdf/1206.5533.pdf

[24] Diederik Kingma,Jimmy Ba. Adam: A Method for Stochastic Optimization arXiv preprint arXiv:1412.6980, 2014. URL https://arxiv.org/pdf/1412.6980v8.pdf.