PRNet: Outdoor Position Recovery for Heterogenous Telco Data by Deep Neural Network

Yige Zhang
Weixiong Rao
yigezhang@tongji.edu.cn
wxrao@tongji.edu.cn
School of Software Engineering
Tongji University
Shanghai, China

Kun Zhang
kunz1@cmu.edu
Department of Philosophy
Carnegie Mellon University
Pittsburgh, United States

Mingxuan Yuan
Jia Zeng
yuan.mingxuan@huawei.com
zeng.jia@huawei.com
Huawei Noah's Ark Lab
Hong Kong, China

ABSTRACT

Recent years have witnessed unprecedented amounts of telecommunication (Telco) data generated by Telco networks. For example, measurement records (MRs) are generated to report the connection states, e.g., received signal strength, between mobile devices and Telco networks. MR data have been widely used to precisely recover outdoor locations of mobile devices for the applications e.g., human mobility, urban planning and traffic forecasting. Existing works using first-order sequence models such as the Hidden Markov Model (HMM) attempt to capture the spatio-temporal locality in underlying mobility patterns for lower localization errors. Such HMM approaches typically assume stable mobility pattern of underlying mobile devices. Yet real MR datasets frequently exhibit heterogeneous mobility patterns due to mixed transportation modes of underlying mobile devices and uneven distribution of the positions associated with MR samples. To address this issue, we propose a deep neural network (DNN)-based position recovery framework, namely PRNet, which can ensemble the power of CNN, sequence model LSTM, and two attention mechanisms to learn local, short- and long-term spatio-temporal dependencies from input MR samples. Extensive evaluation on six datasets collected at three representative areas (core, urban, and suburban areas in Shanghai, China) indicates that PRNet greatly outperforms seven counterparts.

ACM Reference Format:
Yige Zhang, Weixiong Rao, Kun Zhang, Mingxuan Yuan, and Jia Zeng. 2019. PRNet: Outdoor Position Recovery for Heterogenous Telco Data by Deep Neural Network. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357908

1 INTRODUCTION

Recent years have witnessed unprecedented amounts of telecommunication (Telco) data generated by Telco networks. For example, when mobile devices make phone calls or access data services, measurement records (MRs) are generated to report the connection states, e.g., received signal strength of mobile device during each call/session. In a modern urban city, Telco data generated by Telco radio and core equipment account to 2.2TB per day [4]. Massive Telco historical data (e.g., MRs) have been widely used to understand human mobility [2] and facilitate the applications such as urban planning and traffic forecasting [3], due to the unique advantages – the data can be collected cheaply, frequently and on a global scale.

The key of human mobility and the applications above is to capture outdoor positions of mobile users. The position recovery problem, i.e., how to precisely recover outdoor positions of mobile devices from Telco historical data (e.g., MRs), has attracted intensive research interests. A simple approach adopted by Google MyLocation [1] is to approximate outdoor locations by the positions of cellular towers connected by mobile devices. This method suffers from median errors of hundreds and even thousands of meters. More recently, by using the sparse geo-tagged MR data as training data, machine learning-based Telco localization approaches such as Random Forest [29] and Hidden Markov Model (HMM) [9, 12, 15] are exploited to recover outdoor locations of mobile devices with the median errors of tens of meters. In particular, when people holding mobile devices are moving around, the associated locations obviously exhibit spatio-temporal locality. The sequence model-based localization using HMM [9, 12, 15] captures spatio-temporal locality and leads to better localization result than single point-based approaches [29] which independently localize each individual MR sample.

Unfortunately, the HMM approaches above typically assume stable mobility pattern to localize MR samples. For example, they require stable distribution of emission and transition probabilities to precisely estimate the positions of MR samples, and do not work well in real Telco MR data. Frequently real MR data exhibit heterogeneous mobility patterns due to uneven distribution of the positions of MR samples and mixed transportation modes of underlying mobile devices. For example, due to the various timestamp intervals between neighbouring MR samples, the physical distance between two neighbouring MR samples with a time interval of 1 minute is smaller than the one of 10 minutes. Even with the same time interval (e.g., 1 minute), the distance between two neighbouring MR samples in the driving mode is significantly greater than the one in the walking mode. Given such heterogeneous MR samples, the first-order HMM models cannot

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357908

precisely capture spatio-temporal locality and there still exist potentials for much better localization accuracy.

To overcome the challenge above, we propose a hierarchical deep neural network (DNN)-based position recovery framework, namely PRNet. To precisely learn local, short- and long-term spatio-temporal locality from heterogeneous MR samples, PRNet leverages the power of a convolutional neural network (CNN), a recently developed sequence learning model, i.e., long short-term memory cells (LSTM), and attention mechanism. Specifically, PRNet 1) allows the various-length windows of MR samples, such that the CNN and LSTM components have chance to capture spatial locality from the samples within each window, and 2) designs two attention mechanisms for the time-interval between neighbouring MR samples, together with the one between neighbouring MR sequences to capture temporal locality. To this end, PRNet develops a network structure consisting of local and global predictors to merge the captured spatio-temporal locality. The local predictor captures local dependencies among Telco base stations and signal strength within each individual MR sample, and the global recurrent predictor captures the short- and long-term spatio-temporal dependencies from MR sequences to finally generate a smooth trajectory of precisely estimated locations. As a summary, we make the following contributions.

1) To the best of our knowledge, this is the first attempt to leverage the power of the DNN model to automatically learn spatio-temporal knowledge from heterogeneous MR samples. Compared with the existing work on HMM-based localization, PRNet can train a spatio-temporal sequence prediction model from heterogeneous MR data for accurate location recovery.

2) As two key components of PRNet, the local predictor leverages a convolution layer and a recurrent layer to jointly learn local dependencies from each individual MR sample, and the global recurrent predictor, together with two attention mechanisms, extracts the short- and long-term dependencies from a series of sequential MR samples.

3) We evaluate PRNet against state-of-the-art counterparts by using six datasets collected on three representative (core, urban and rural) areas in Shanghai, China. Our evaluation results indicate that PRNet significantly outperforms these counterparts.

2 PRELIMINARIES

Measurement Record (MR) Data: MR data measures the connection states between mobile devices and neighbouring base stations. Table 1 gives an example of 2G GSM MR collected by an Android phone. This sample contains a unique number (known as IMSI, International Mobile Subscriber Identity), connection time stamp (MRTime), up to 7 nearby base stations (RNCID and CellID), signal measurements such as AsuLevel, SignalLevel, and RSSI (radio signal strength indicator). The 7 base stations are frequently sorted by descending order in signal level and strength. Due to the strongest signal, the base station with the order index 1 (RNCID_1 and CellID_1) is typically selected as the primary serving base station to provide communication and data services for mobile devices. When a mobile device is moving out of the signal coverage range of a primary serving base station, the *handoff* between base stations occurs to re-select a new primary serving base station for the mobile device.

Telco Location Recovery: Existing works on Telco position recovery are typically divided into three categories. Firstly, *measurement-based methods*, e.g., triangulation-based approaches, localize mobile devices by estimating absolute point-to-point distance or angles [14, 20, 22]. Triangulation-based approaches usually do not work well for 4G LTE MR data, which frequently contain signal strengths of one or at most two cells. Yet triangulation approaches require signal strength regarding at least three and ideally four or more cells.

Secondly, *fingerprinting-based methods* [10, 11, 13] usually have better performance than the measurement-based methods above by dividing an area of interest into small grids and representing each grid by an associated fingerprint [11]. NBL [13], a recently proposed work, assumed that signal strengths of each neighbouring cell tower in the grid follow a Gaussian distribution. The online stage next adopts either Maximum Likelihood Estimation (MLE) or Weighted Average (WA) to localize mobile devices.

Lastly, some recent works [10, 13, 18, 27, 29] adopt *machine learning (ML) techniques* such as Random Forest to build a localization model to maintain the correlations between the features extracted from MR samples and associated locations (e.g., GPS coordinates). The predicted locations could be either spatial regions (grid cells) or numeric GSP coordinates. ML techniques then train the corresponding multi-classifiers [10, 13, 18] or regression models [29]. Recently, a deep neural network-based outdoor cellular localization system, namely DeepLoc [18], has been proposed. DeepLoc mainly utilizes a data augmentor to handle data noise issue and to provide more training samples, and trains a deep learning model using the augmented samples for better localization result.

Note that ML techniques significantly differ from measurement- and fingerprinting-based methods in terms of application scenarios. That is, ML techniques work as Telco *backend* approaches on *historical* Telco big data that are maintained by Telco operators. ML techniques usually exhibit better results because ML techniques can leverage the rich information from various Telco data including (geo-tagged) MR data and configuration parameters (e.g., GPS coordinates) of base stations, Web log data and etc. Instead, measurement and fingerprinting-based methods usually work on *frontend* mobile devices as active localization and may not use the rich data as ML techniques do.

3 PROBLEM SETTING

Suppose that Telco operators have maintained a historical MR database. MR samples usually do not contain the accurate locations of mobile devices (identified by IMSI), and we expect to annotate MR samples with the accurate locations of these mobile devices. There are various ways to acquire the locations. For example, when mobile users are using vehicle navigation services and switching on GPS receivers in the mobile devices, the GPS coordinates of mobile devices are embedded in the URLs of mobile web logs. By extracting GPS coordinates from such URLs, we establish linkage between the extracted GPS coordinates and MR samples with help

MRTime	2018/4/23 9:20	IMSI	xxx	SRNCID	6188	BestCellID	26051	LCS BIT	300
RNCID_1	6188	CellID_1	26051	AsuLevel_1	27	SignalLevel_1	4	RSSI_1	-74.5
RNCID_2	6188	CellID_2	27394	AsuLevel_2	10	SignalLevel_2	3	RSSI_2	-84.88
RNCID_3	6188	CellID_3	27377	AsuLevel_3	18	SignalLevel_3	4	RSSI_3	-85.13
RNCID_4	6188	CellID_4	27378	AsuLevel_4	12	SignalLevel_4	4	RSSI_4	-85.87
RNCID_5	6182	CellID_5	41139	AsuLevel_5	8	SignalLevel_5	3	RSSI_5	-88.88
RNCID_6	6188	CellID_6	27393	AsuLevel_6	9	SignalLevel_6	3	RSSI_6	-90.22
RNCID_7	6182	CellID_7	44754	AsuLevel_7	9	SignalLevel_7	3	RSSI_7	-95

Table 1: An 2G GSM MR Sample

of IMSI and timestamp. In this way, MR samples are annotated by the linked GPS coordinates.

Nevertheless, mobile users frequently switch off GPS receivers in mobile devices e.g., for energy saving. Mobile web logs contain rather sparse GPS coordinates. Thus, the majority of Telco MR data cannot be annotated with the associated GPS coordinates. To tackle this issue, we could first train a machine learning model by using the sparse geo-tagged MR samples as training data and next recover the outdoor locations for non-geo-tagged MR samples (as testing data).

Recall that single-point-based location recovery [29] may not capture spatio-temporal locality of underlying mobile devices. We thus expect to recover a trajectory of the locations predicted from a sequence of MR samples.

Definition 1. ***[MR Database $\mathcal{D}$]*** *A database $\mathcal{D}$ of MR samples is organized as follows. We first group the MR samples by IMSI and next sort the samples in each group by timestamp. In this way, each IMSI is with a series of sorted MR samples.*

Suppose that every MR sample in $\mathcal{D}$ is with an associated GPS position. Then for each ISMI, we have a series of sorted MR samples and an associated trajectory of GPS positions. We are now ready to define the following problem.

Problem 1. ***[Position Recovery Problem]*** *Given a database $\mathcal{D}$, by training a machine learning-based position recovery model $\mathbb{R}$ mapping the MR series of every ISMI to the associated GPS trajectory, we predict the GPS trajectory for an input MR series.*

A baseline approach to address Problem 1 is to exploit a sequence model such as the popular RNN model or its variant LSTM. For example, we might divide every MR series into multiple fixed-length windows of MR samples and feed these windows of MR samples into the LSTM structure for trajectory recovery. However, due to the following challenges caused by heterogeneous MR samples, the baseline approach does not work well to capture spatio-temporal locality, thus leading to high errors.

1) *Mixed transportation modes*: The simple assumption that the mobile device regarding a given MR series always moves by a certain transportation mode (e.g., walking) clearly does not hold. Depending on transportation mode, the spatial distance between neighbouring MR samples significantly varies.

2) *Irregular MR sampling rate*: Practically it is highly possible that the time intervals of some neighbouring MR samples within a given MR series are very short (e.g., 1 minute) and yet those of other samples are rather long (e.g., 10 minutes). Thus, assuming fixed time interval between neighbouring MR samples does not work.

3) *Uneven density of deployed base stations*: Dense base stations are frequently deployed in urban areas and yet spare ones in rural areas. Consequently, the neighbouring MR samples within the MR series located in urban and rural areas, even with exactly the same spatial distance and time interval, could exhibit very different handoff behavior between base stations.

Given the challenges above, neither the baseline approach using the basic RNN / LSTM model nor the traditional HMM model is expected to precisely capture the underlying mobility pattern in general situations for low localization errors. To properly address those issues, the proposed PRNet 1) allows the various-length windows of MR samples, such that the local predictor in PRNet can capture spatial locality that is exhibited by the MR samples within each window, 2) designs the two attention mechanisms for time intervals and MR (sub)sequences to capture temporal locality, and 3) develops a network structure consisting of local and global predictors to merge the captured spatio-temporal locality.

In the rest of this section, we first introduce MR (sub)sequences which allow for various length MR samples, and next describe the design of PRNet which takes MR (sub)sequences as the input.

3.1 MR (Sub)sequence

To make sure that the proposed local predictor in PRNet can capture the spatial locality from the various-length MR samples within each window, we exploit the handoff between base stations as follows. Let us consider the coverage range of a certain base station bs, e.g., a circle with the radius around 1000 meters. The MR samples using bs as the primary serving base station exhibit spatial locality because these samples are located within the circle. Thus, based on the base stations in MR samples, we define the following MR (sub)sequences.

Definition 2. ***[MR Sequence $\mathcal{S}$]*** *Given a series of sorted MR samples regarding a certain ISMI, we divide the MR series into multiple sequences by primary serving base stations (*RNCID_1 *and* CellID_1*). The MR samples in each sequence $\mathcal{S}$, sorted by timestamp, share the same primary serving base station and IMSI.*

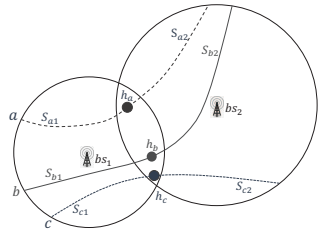


Figure 1: Illustration of 6 MR sequences

Figure 1 illustrates 6 MR sequences $S_{a1}, S_{a2}, ..., S_{c1}, S_{c2}$ generated from 3 MR series identified by the 3 mobile devices a, b, c across the coverage range of two primary serving base stations bs_1 and bs_2. For a mobile device say a, the handoff point h_a divides its MR series into two sequences S_{a1} and S_{a2}.

Beyond MR sequence, we define the following MR subsequence:

Definition 3. ***[MR Subsequence** S**]*** *Given a certain MR sequence $\mathcal{S}$, we divide $\mathcal{S}$ into multiple subsequences S by the first non-serving base station (RNCID_2 and CellID_2). The MR samples in each subsequence share exactly the same non-serving base station (RNCID_2 and CellID_2), primary serving base station and IMSI.*

The amount of MR samples within each (sub)sequence varies, depending upon the coverage radius of a base station, sampling rate, and transportation mode. In case that a certain MR (sub)sequence contains very sparse samples, we could adaptively merge multiple neighbouring MR (sub)sequences into a single one which contains at least τ samples. Since these neighbouring MR (sub)sequences still exhibit (though less) spatial locality, it makes sense for PRNet to capture the corresponding spatial locality from these MR (sub)sequences.

4 DESIGN OF PRNET

In this section, we give the detail of PRNet. For convenience, we denote scalars by lowercase letters e.g., a, vectors by bold lowercase letters e.g., $\boldsymbol{a}$, matrices by bold upper-case letters e.g., $\boldsymbol{A}$, and tensors by bold upper-case letters e.g., $\mathbf{A}$.

4.1 Framework Overview

In this section, we first give the data model in PRNet. Specifically, we represent each MR sample by a matrix $\boldsymbol{X} \in \mathbb{R}^{F \times N}$, where N is the number of base stations in this MR sample and F is the number of MR features of each base station. For the 2G MR sample in Table 1, we have $N = 7$ base stations (from the primary serving one to the 7-th station) and $F = 7$ features (i.e., RNCID, CellID, latitude and longitude of the base station identified by RNCID / CellID, ASULevel, SignalLevel, and RSSI).

Next, we represent each MR sequence $\mathcal{S}$ by an associated sequence of MR feature matrices $\mathbf{X} = \{\boldsymbol{X}_{1,1}, ..., \boldsymbol{X}_{1,l_1}, ..., \boldsymbol{X}_{q,l_q}\}$, where $\boldsymbol{X}_{i,j} \in \mathbb{R}^{F \times N}$ is the MR feature matrix of the j-th MR sample in the i-th subsequence S_i with $1 \le i \le q$ and $1 \le j \le l_i$ where q is the number of MR subsequences within $\mathcal{S}$ and l_i is the amount of MR samples in a subsequence S_i. Meanwhile, we represent a sequence of positions $\boldsymbol{Y} = \{\boldsymbol{y}_{1,1}, ..., \boldsymbol{y}_{1,l_1}, ..., \boldsymbol{y}_{q,l_q}\}$ where $\boldsymbol{y}_{i,j}$, i.e., the GPS position of $\boldsymbol{X}_{i,j}$, is a 2×1 vector due to the two dimensional GPS latitude and longitude coordinates.

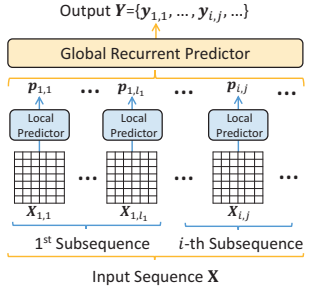


Figure 2: Framework overview of PRNet

Given a database $\mathcal{D}$ containing the input sequences $\mathbf{X}$ and associated GPS trajectory sequences $\boldsymbol{Y}$, PRNet trains a position recovery model $\mathbb{R}$, which then predicts the output sequence $\boldsymbol{Y}'$ of a testing sequence $\mathbf{X}'$. Figure 2 gives the overview of PRNet, consisting of two main components.

1) **Local Single-Point Predictor** takes each individual matrix $\boldsymbol{X}_{i,j}$ as the input, and generates a corresponding hidden vector $\boldsymbol{p}_{i,j}$ as the output. The predictor is composed of a convolution layer and a recurrent layer to automatically capture the local dependencies from the MR features in $\boldsymbol{X}_{i,j}$.

2) **Global Recurrent Predictor** takes a sequence of generated hidden vectors $\boldsymbol{P} = \{\boldsymbol{p}_{1,1}, ..., \boldsymbol{p}_{1,l_1}, ..., \boldsymbol{p}_{q,l_q}\}$ as the input and generates a sequence of predicted positions $\boldsymbol{Y}$. This predictor consists of three layers: a bottom recurrent layer to learn short-term dependencies within each subsequence, an upper recurrent layer to capture the long-term dependencies among subsequences, and an output layer to mix the short- and long-term dependencies in order to generate the final position prediction.

4.2 Local Predictor

Each MR sample contains the signal measurements of up to 7 base stations. Thus, as shown in Figure 3, the convolution layer in the local predictor captures the local dependencies among the F MR features from the input MR feature matrix $\boldsymbol{X}_{i,j}$. Meanwhile, the MR sample can be alternatively treated as a sequence of the N base stations sorted by the associated signal measurements. Thus, the recurrent layer captures the local dependencies from the sequence.

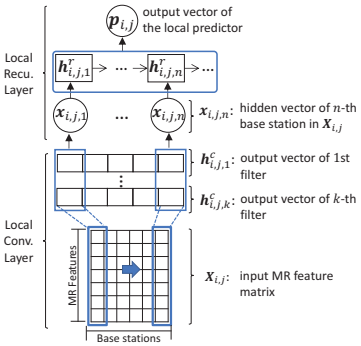


Figure 3: Network structure of local predictor

4.2.1 Local Convolution Layer. Given an input $F \times N$ matrix $\boldsymbol{X}_{i,j}$, the convolution layer captures the local dependencies between MR features. Thus, the convolution layer adopts multiple one-dimension filters with size $F \times 1$ where the height of each filter is equal to the number F of features. Since the size of the convolution filter is consistent with the feature vector of a certain base station in $\boldsymbol{X}_{i,j}$, the convolution operation can extract the local dependencies among multiple features of the base station. The output vector of the k-th filter throughout the input matrix $\boldsymbol{X}_{i,j}$ is:

$$\boldsymbol{h}^c_{i,j,k} = RELU(\boldsymbol{W}^c_k \circ \boldsymbol{X}_{i,j} + \boldsymbol{b}^c_k), \quad (1)$$

where $\circ$ denotes the convolution operation and $RELU(\cdot)$ is the activation function. The superscript c indicates the convolution layer. We set the stride of convolution operation by 1 to ensure that each output vector $\boldsymbol{h}^c_{i,j,k}$ has the size equal to N.

Given K filters, the output of the local convolution layer is a $K \times N$ matrix. For the n-th base station ($1 \le n \le N$) within $\boldsymbol{X}_{i,j}$, the $K \times N$ output matrix has an associated row vector $\boldsymbol{x}_{i,j,n} \in \mathbb{R}^{K \times 1}$, treated as the latent feature vector of the n-th base station.

4.2.2 Local Recurrent Layer. Given a $K \times N$ matrix above, the local recurrent layer treats it as a sequence of N latent feature vectors. The intuition is that the order of these latent feature vectors indicates the inherent correlations among the N base stations in an input MR sample. Thus, this local recurrent layer extracts the local dependencies from the latent feature vectors regarding N base stations. We implement the local recurrent layer by LSTM [7] and compute the hidden state of LSTM cells for the latent vector $\boldsymbol{x}_{i,j,n}$ as follows.

$$\begin{aligned}
\boldsymbol{z}^r_{i,j,n} &= \tanh(\boldsymbol{W}_z{}^r[\boldsymbol{h}^r_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^r_z),\\
\boldsymbol{f}^r_{i,j,n} &= \sigma(\boldsymbol{W}_f{}^r[\boldsymbol{h}^r_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^r_f),\\
\boldsymbol{g}^r_{i,j,n} &= \sigma(\boldsymbol{W}_g{}^r[\boldsymbol{h}^r_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^r_g),\\
\boldsymbol{c}^r_{i,j,n} &= \boldsymbol{f}^r_{i,j,n} * \boldsymbol{c}^r_{i,j,n-1} + \boldsymbol{g}^r_{i,j,n} * \boldsymbol{z}^r_{i,j,n},\\
\boldsymbol{o}^r_{i,j,n} &= \sigma(\boldsymbol{W}_o{}^r[\boldsymbol{h}^r_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^r_o),\\
\boldsymbol{h}^r_{i,j,n} &= \boldsymbol{o}^r_{i,j,n} * \tanh \boldsymbol{c}^r_{i,j,n},
\end{aligned} \tag{2}$$

where $*$ denotes the element-wise multiplication and $\boldsymbol{h}_{i,j,n}$ is the hidden state of the n-th base station in $\boldsymbol{X}_{i,j}$. The superscript r denotes a local recurrent layer. The output of this layer, $\boldsymbol{p}_{i,j} \in \mathbb{R}^{(N \times d_l) \times 1}$, is the concatenation result of all hidden states, where d_l denotes the number of hidden units in this local recurrent layer.

By the local predictor, PRNet encodes each individual MR feature matrix $\boldsymbol{X}_{i,j}$ into a hidden feature vector $\boldsymbol{p}_{i,j}$. When an entire matrix sequence $\mathbf{X}$ is processed, the local predictor generates a sequence $\boldsymbol{P} = \{\boldsymbol{p}_{1,1}, ..., \boldsymbol{p}_{i,j}, ..., \boldsymbol{p}_{q,l_q}\}$, which is fed into the global recurrent predictor to generate the final trajectory.

4.3 Global Recurrent Predictor

Figure 4 gives the neural network structure of our global recurrent predictor. Inspired by the hierarchical deep neural network for document classification problem [25], the global recurrent predictor first employs a *bottom recurrent layer* with time interval attention to extract the latent short-term dependencies among MR samples within a subsequence. An *upper recurrent layer* is next exploited to learn the long-term dependencies among MR subsequences to generate subsequence attention. Finally, an *output layer* merges the hidden state of bottom recurrent layer and generated subsequence attention of upper recurrent layer to produce the final output.

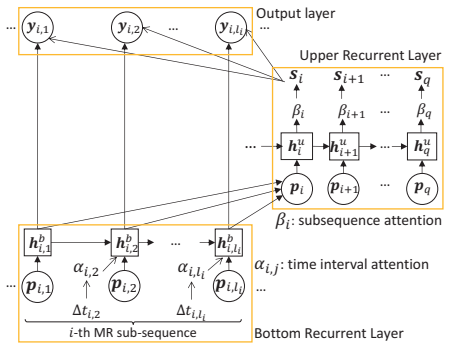


Figure 4: Network Structure of Global Recurrent Predictor

4.3.1 Bottom Recurrent Layer. The bottom recurrent layer adopts LSTM cells to capture the short-term dependencies among the hidden vectors $\boldsymbol{p}_{i,1}, ..., \boldsymbol{p}_{i,l_i}$ within a certain subsequence S_i. However, a standard LSTM model ignores the difference of time intervals between the neighbouring cells. The length of time interval indicates the relevance from the previous cell to the current one. For instance, if the time interval increases, the contribution of previous cell becomes weak. Thus, we utilize the time-interval attention mechanism to address this issue.

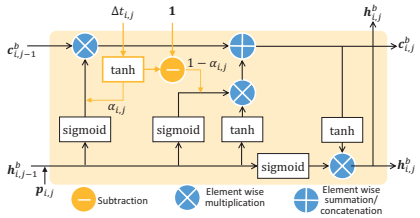


Figure 5: Time-interval Attention in LSTM Cells

Time-interval attention: Figure 5 gives the LSTM cell structure with time interval attention. Specially, the attention acts on two LSTM gates: *1)* forget gate and *2)* input gate, respectively. We construct the time interval attention via a liner perceptron by referring to the time interval between the current state and the previous one with:

$$a_{i,j} = \tanh(w^a \Delta t_{i,j} + b^a), \tag{3}$$

where w^a and b^a are learnable parameters and $\Delta t_{i,j}$ is the timestamp difference between the current MR sample w.r.t $\boldsymbol{p}_{i,j}$ and the previous one. We denote the time interval attention weight by $a_{i,j}$ to update the forget and input gates of LSTM cell as:

$$\begin{aligned}
\boldsymbol{z}^b_{i,j,n} &= \tanh(\boldsymbol{W}_z{}^b[\boldsymbol{h}^b_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^b_z),\\
\boldsymbol{f}^b_{i,j,n} &= \sigma(\boldsymbol{W}_f{}^b[\boldsymbol{h}^b_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^b_f) \cdot a_{i,j}\\
\boldsymbol{g}^b_{i,j,n} &= \sigma(\boldsymbol{W}_g{}^b[\boldsymbol{h}^b_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^b_g) \cdot (1 - a_{i,j}),\\
\boldsymbol{c}^b_{i,j,n} &= \boldsymbol{f}^b_{i,j,n} * \boldsymbol{c}^b_{i,j,n-1} + \boldsymbol{g}^b_{i,j,n} * \boldsymbol{z}^b_{i,j,n},\\
\boldsymbol{o}^b_{i,j,n} &= \sigma(\boldsymbol{W}_o{}^b[\boldsymbol{h}^b_{i,j,n-1}; \boldsymbol{x}_{i,j,n}] + \boldsymbol{b}^b_o),\\
\boldsymbol{h}^b_{i,j,n} &= \boldsymbol{o}^b_{i,j,n} * \tanh \boldsymbol{c}^b_{i,j,n},
\end{aligned} \tag{4}$$

where superscript b indicates the bottom recurrent layer. The time attention $a_{i,j}$ first acts on the forget gate and models a temporal decay to discard the information from the previous cell state. Next, since the latest state mainly determines the output of the current input, the attention $(1 - a_{i,j})$ is thus applied to the input gate.

4.3.2 Upper Recurrent Layer. After the bottom recurrent layer has extracted the short-term temporal dependencies among the MR samples within a subsequence, the upper recurrent layer next captures the long-term dependencies between MR subsequences with the following input:

$$\boldsymbol{p}_i = \sum_j \boldsymbol{h}^b_{i,j}, \tag{5}$$

where $\boldsymbol{h}^b_{i,j}$ denotes the j-th hidden state of the i-th subsequence acquired from the bottom recurrent layer. Since the upper recurrent layer is to capture the correlations among subsequences, the input of this layer, $\boldsymbol{p}_i$, needs to consider the latent characteristics of all elements in the i-th subsequence. Thus, we compute the input $\boldsymbol{p}_i$ by the sum of the hidden states in the i-th subsequence.

Subsequence attention: In the upper recurrent layer, we again leverage the LSTM cells to capture the long-term dependency by a

subsequence attention mechanism to adaptively select meaningful hidden states (subsequences) across all subsequences ($i = 1, ..., q$). Specifically, the attention weight of i-th hidden state $\boldsymbol{h}_i^u$ regarding the i-th subsequence (denoted by $\boldsymbol{p}_i$) is computed as:

$$u_i = (\boldsymbol{v}^u)^\top \tanh(\boldsymbol{W}^u \boldsymbol{h}_i^u + \boldsymbol{b}^u),$$
$$\beta_i = \frac{\exp(u_i)}{\sum_{m=1}^{q} \exp(u_m)}, \tag{6}$$

where $\boldsymbol{v}^u$, $\boldsymbol{W}^u$ and $\boldsymbol{b}^u$ are learnable parameters. The attention weight β_i represents the importance of the i-th hidden state $\boldsymbol{h}_i^u$ for the prediction, where superscript u refers to the upper recurrent layer. Based on the weight β_i, the upper recurrent layer generates a weighted context vector $\boldsymbol{s}_i = \beta_i \boldsymbol{h}_i^u$ as the output.

4.3.3 Output Layer. The output layer is a dense layer to merge the outputs of the bottom and upper recurrent layer. The output layer takes the hidden state of bottom layer at time step j of the i-th subsequence (denoted by $\boldsymbol{h}_{i,j}^b$) and the context vector of upper layer of the i-th subsequence (denoted by $\boldsymbol{s}_i$) as input. The final output can be formulated as:

$$\boldsymbol{y}_{i,j} = \tanh(\boldsymbol{W}^b \boldsymbol{h}_{i,j}^u + \boldsymbol{W}^u \boldsymbol{s}_i + \boldsymbol{b}), \tag{7}$$

4.4 PRNet Training

To minimize the mean squared error between the predicted trajectories and their corresponding ground truth, the objective function of PRNet is formulated as:

$$\min_{\theta} \sum ||Y' - Y^{true}||^2 \tag{8}$$

where θ denotes the parameters in PRNet, Y' is the predicted trajectory which contains a series of location coordinates, and Y^{true} is the ground truth trajectory. In our experiments, we set the number of hidden units in local convolution, local recurrent, bottom recurrent and upper recurrent layer by 16, 256, 100, and 200, respectively. The sequence-based position recovery becomes a regression problem, and Stochastic Gradient Decent (SGD) optimizer can solve this problem.

5 EXPERIMENTS

5.1 Experimental Setup

	Jiading	Siping	Xuhui
Num. of samples (2G/4G)	17354/12245	6723/4953	14680/10455
Route len (2G/4G) in km	96.5/60.3	24.6/15.5	29.3/15.9
Sampling rate (sec)	3	3	2
Area size (km^2)	1.67	0.862	0.57
% of MRs with walking/cycling/driving	64.7/25.2/14.1	63.8/36.2/0.0	58.5/0.0/41.5
Serving BSs density (2G/4G)	5.72/8.45	7.96/10.35	8.45/12.67

Table 2: Statistics of Used Data Sets

5.1.1 Data Sets. We evaluate the performance of PRNet using the data sets collected in three representative areas in Shanghai, China: a core business area *Xuhui*, an urban area *Siping* and a rural area *Jiading*. The geographical distances between Jiading and Siping, between Jiading and Xuhui, and between Siping and Xuhui are around 31 km, 37 km and 15 km, respectively. The data sets (including both MR samples and GPS location labels) in Xuhui were provided by one of the largest Telco operators in China, and the data sets for Jiading and Siping were collected by our developed Android mobile app. When mobile users are moving around in outdoor road networks to collect MR samples, we meanwhile switch on GPS receivers on Android mobile phones to collect current GPS coordinates. Since the collected GPS coordinates may contain noises, we thus employ the map-matching technique [28] to mitigate the effect of noises. To protect the users' privacy, all sensitive information such as IMSI have been anonymized.

Table 2 summarizes the used data sets. In each area, we have two MR data sets collected from 2G GSM and 4G LTE networks, respectively. Each sample in the data sets includes one MR record and the associated GPS coordinates. Due to the limitations of the Android APIs in the 4G networks, each 4G MR sample Jiading and Siping contains only one (serving) base station without the information about other neighboring base stations. Yet each 4G MR sample in Xuhui still contains up to 7 base stations.

The data sets contain heterogenous MR samples due to mixed transportation modes and uncertain timestamp intervals between neighbouring samples. In particular, though we set a fixed sampling rate to collect MR data, the timestamp intervals between neighbouring samples vary significantly. For example, in the Jiading 2G data set, the timestamp intervals vary from 1 to 125 seconds, mainly due to: (1) the uncertain delay of Android threads scheduled by Android OS to collect MR samples and (2) noisy MR samples (e.g., those samples having empty or zero signal measurements or missed base station IDs) which we have to clean. Given the heterogenous samples, we will study the effect of transportation modes and sampling rates in Section 5.3.

Besides, we acquire the GPS longitude and latitude coordinates of every base station provided by Telco operators, and use them as additional features of MR samples. However, the parameters such as antenna height and angle are unavailable due to the limited information provided by the database. We believe that these parameters, if available, will further improve PRNet.

Counterpart	Description	Approach
NBL	Recent fingerprinting method [13]	Single Point
DeepLoc	3-layer neural network [18]	Single Point
RaF	1-layer Random Forest regression [29]	Single Point
CCR	2-layer Random Forest regression [29]	Implicit Sequence
HMM	HMM + particle filtering [15]	Sequence
SeqtoSeq	a LSTM-based seq. to seq. model [19]	sequence
ConvLSTM	a convolutional LSTM [17]	Sequence
PRNet	proposed hierarchical neural network	Sequence

Table 3: Counterparts

5.1.2 Counterparts. We compare PRNet against 7 counterparts in Table 3 from the following aspects:

1) Depending upon the location recovery result, the counterparts are either single point- (NBL [13], RaF [29], CCR [29] and a very recent work DeepLoc [18]), or sequence-based approaches (HMM [15], SeqtoSeq [19], ConvLSTM [17] and PRNet). In [29], the location recovery model can be either only a single-layer Random Forest (RaF), or a two-layer Random Forests (CCR) which can be treated as an implicit sequence-based approach due to the contextual features acquired from the predicted result of the 1st layer.

*2)*In terms of the used models, the counterparts can be either the fingerprinting-based (NBL), or traditional machine learning-based (RaF, CCR and HMM), or DNN-based models (SeqtoSeq, ConvLSTM, DeepLoc and PRNet).

Since our datasets contain a small amount of MR samples, we follow the previous work [6, 16] to adopt the k-fold ($k = 10$) cross

validation by choosing 80% training and 20% testing data from each data set to avoid over-fitting. We compute the prediction error by the Euclidian distance between recovered locations and ground truth (i.e., the real GPS coordinates of MR samples), we choose median error, mean error, and top 90% error (by sorting prediction errors in ascending order) as evaluation metrics.

Parameter	Range	Default Value
Base station density	25% - 100%	100%
Timestamp Interval (sec)	3–120	3
Number of training epochs	100 - 1400	1000
Transportation mode	mixed, walk, bike, drive	mixed
Learning rate	0.0001, 0.0005, 0.001, step-decay	step-decay

Table 4: Key Parameters

Table 4 lists the parameters used in our experiments. We use default values in the baseline experiment, and vary the values within the allowable ranges for sensitivity study.

5.2 Baseline Study

In Table 5, we compare the prediction errors (median, mean, and 90% errors) of the eight approaches on the six datasets from three areas (Jiading, Siping, and Xuhui). In addition, for the single-point approaches such as CCR, NBL, and DeepLoc, we particularly perform a post-processing step on their predicted locations by using the particle filtering (PF) technique, namely CCR_F, etc. Following [18], we set the grid size as 100 meters in NBL and DeepLoc for fairness. Following [15], the parameters of PF are as follows: 1) 500 particles, 2) 100 times of recurrent processes, and 3) Gaussian distribution for system and observation noises. Moreover, the map-matching technique [28] is utilized to project the predicted locations by NBL, DeepLoc, CCR, HMM, and PRNet onto the road networks, and name them by NBL_M, etc. We have the following findings from Table 5.

1) PRNet offers the least error among these eight approaches on six MR datasets. For example, in Jiading 2G data set, PRNet reduces the median error by 55.0% compared with the recent outdoor Telco position recovery approach, DeepLoc. It is mainly because DeepLoc is a single point-based prediction method, which does not capture contextual dependencies. In addition, PRNet outperforms the two deep neural network-based approaches (i.e. SeqtoSeq and ConvLSTM). Such results indicate that PRNet outperforms both the state-of-the-arts and alternative DNN-based approaches.

2) We compare the top-90% errors among the eight approaches. Still on Jiading 2G dataset, PRNet outperforms CCR with 55.6% lower top-90% error. Similar results apply to other datasets. These numbers indicate that PRNet has leveraged the power of hierarchical deep neural network, sequence model, and attention mechanism together to effectively mitigate the outliers among the predicted locations.

3) We are interested in the result of two post-processing techniques. First, the particle filtering (PF)-based methods NBL_F, DeepLoc_F, and CCR_F lead to slightly better results than the original ones. However, PRNet still outperforms these PF-based results mainly because particle filtering can correct individual outliers but not multiple continuous outliers which could entirely shift away from ground truth locations. Second, the map matching-based results leverage road network constraints and lead to much lower errors than particle filters. Nevertheless, most of these map matching-based results still cannot compete with PRNet. It is mainly because the map-matching technique cannot fully eliminate the horizontal shift along the road, which can be visualized in the following section. Not surprisingly, PRNet_M outperforms all the other approaches.

4) We compare the results of the four sequence-based approaches against 4 single-point-based ones. For example, a simple sequence model, HMM, still outperforms DeepLoc by 31.0% smaller 90% error on Jiading 2G data set. In addition, though CCR and RaF are both Random Forest-based approaches, CCR leverages the contextual features such as moving speed and leads to better result than RaF by 10.9% smaller median error. Such results verify that the sequence models, no matter they implicit or explicit use of contextual features, could lead to better results than the original single-point-based approaches.

5) Among the four deep neural network-based approaches, ConvLSTM suffers the highest errors. It might be mainly because ConvLSTM is typically used to solve flow prediction (e.g., precipitation nowcasting [17] and traffic accident [26]) for the time series data involving explicit spatiotemporal information. However, MR samples do not contain accurate locations, which are just the objective of our model (note that the GPS locations of base stations are used as additional features, but not the locations of MR samples). Thus, ConvLSTM may not properly solve our problem though ConvLSTM also utilizes the power of both CNN and LSTM. SeqtoSeq is slightly better than ConvLSTM, but still cannot compete with PRNet.

6) Table 5 is consistent with results reported in the previous works CCR and NBL: for the same area e.g., Xuhui, 4G MR samples typically lead to better result than 2G samples; and for the same Telco networks, the results of the Xuhui 2G dataset are better than those of Siping and Jiading 2G datasets. We have the finding that higher base station deployment density in 4G networks and core areas (see Table 2) can achieve better result than those in 2G and rural areas. We also note that due to only one base station per MR sample in Jiading and Siping 4G datasets, their errors are even higher than those 2G datasets. Such result verifies the importance of multiple base stations for precise location recovery.

5.3 Sensitivity Study

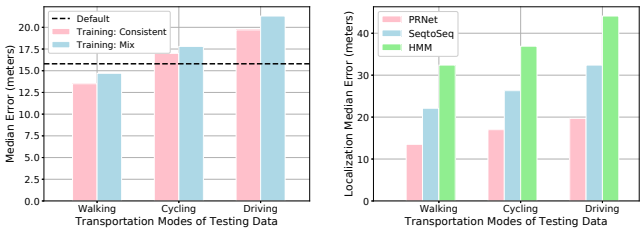
Due to space limitation, we choose the Jiading 2G dataset (with the most data samples and largest area) for sensitivity study.

(1) Transportation Mode: Due to the mixed transportation modes, we are interested in how these modes affect the performance of PRNet on the (training and test) datasets with various modes. Specifically, for the Jiading 2G dataset, we divide MR samples by their transportation modes (walking, cycling and driving). We consider two options to train PRNet by 1) the entire training data with mixed transportation modes or 2) only those training MR sample with a certain mode. Similarly, based on the trained PRNet, we predict the positions of 1) those testing data with mixed transportation modes or 2) those with a certain mode.

Figure 6(a) gives the prediction errors of three modes (each of which is with two optional training datasets). In addition, the black dotted line indicates the localization median error for training/test data both with mixed transportation modes. From this figure, we

Dataset	Jiading(2G)			Jiading(4G)			Siping(2G)			Siping(4G)			Xuhui(2G)			Xuhui(4G)		
	Median	Mean	90%	Median	Mean	90%	Median	Mean	90%	Median	Mean	90%	Median	Mean	90%	Median	Mean	90%
NBL	53.4	67.2	300.9	59.7	72.3	318.6	42.8	63.0	298.3	43.2	64.9	256.7	45.9	59.0	240.7	32.2	52.7	191.2
NBL_F	40.8	59.2	214.7	48.6	60.7	223.8	37.4	56.1	209.3	36.7	57.2	209.3	37.2	51.4	186.5	29.3	48.5	171.3
NBL_M	38.9	50.7	165.8	40.1	54.2	169.5	30.5	45.1	141.8	31.1	46.6	140.6	32.3	45.6	155.1	30.6	44.8	142.9
DeepLoc	35.1	47.6	250.3	40.2	53.9	280.6	33.5	44.7	219.9	38.7	49.6	267.5	31.2	40.5	210.5	27.4	40.1	211.9
DeepLoc_F	31.4	43.5	237.6	35.6	46.5	240.3	29.2	41.5	200.4	33.3	43.2	224.8	27.5	36.9	180.5	24.3	38.7	166.3
DeepLoc_M	21.6	34.4	165.3	29.8	40.7	197.3	20.4	32.7	169.3	26.9	37.4	209.3	21.0	31.5	169.7	19.3	28.5	151.7
RaF	38.3	48.3	168.9	38.5	47.2	158.9	41.1	47.3	158.8	35.1	44.4	105.8	35.6	43.4	137.7	32.2	41.5	104.4
CCR	34.1	43.2	142.3	30.2	44.5	145.9	37.5	42.8	139.5	29.5	40.6	101.2	30.0	40.2	125.2	20.0	34.1	98.3
CCR_F	29.3	40.8	130.7	27.7	41.2	136.3	30.1	39.6	128.4	24.5	37.4	95.6	25.3	34.2	117.3	18.6	32.1	91.9
CCR_M	22.1	36.6	99.1	17.8	32.9	105.5	23.8	33.5	93.9	16.8	32.1	79.9	18.4	34.5	92.6	13.6	28.4	86.7
HMM	36.5	52.3	172.8	42.1	53.6	188.4	37.2	51.4	160.3	30.7	48.2	145.6	35.3	42.7	148.0	28.6	39.1	129.5
HMM_M	22.3	41.7	109.2	24.6	40.3	118.3	25.0	36.6	121.0	19.7	35.1	99.3	21.2	34.8	113.4	18.3	31.9	103.5
SeqtoSeq	25.4	50.6	85.3	24.2	50.1	81.7	23.4	48.9	84.9	22.7	47.3	83.5	22.5	43.2	84.7	21.9	42.9	81.8
ConvLSTM	28.5	59.3	129.3	27.3	58.1	124.5	27.7	57.3	120.4	25.6	55.7	117.2	25.3	52.5	113.2	24.4	50.6	107.8
PRNet	**15.8**	**37.8**	**63.2**	**18.4**	40.6	66.5	**15.3**	**34.2**	**60.2**	17.4	37.7	63.4	**15.7**	**34.4**	**62.5**	**13.7**	**31.2**	**59.3**
PRNet_M	14.1	34.1	60.2	14.7	34.8	62.4	13.7	31.5	58.3	14.9	33.7	60.6	14.2	32.8	60.3	13.1	29.9	57.5

Table 5: Baseline Experiment: Localization Errors of Eight Approaches



(a) Training/Testing Data (b) Sequence Model

Figure 6: Effect of Transportation Modes

find that the walking mode leads to the best prediction accuracy. It is mainly because the MR samples collected by walking exhibit higher spatial locality compared to cycling and driving, i.e., more MR samples within every MR (sub)sequence. Secondly, the consistent transportation modes between training and test datasets (i.e., the "consistent" legend in Figure 6(a)) lead to better performance than the mixed modes. It is reasonable because the homogeneous MR samples with the same transportation mode are helpful to better capture stable spatiotemporal dependencies.

In addition, Figure 6(b) studies how three approaches (PRNet, SeqtoSeq, and HMM) perform under various transportation modes. Among all transportation models, PRNet still performs best. In particular, the errors of both SeqtoSeq and HMM grow significantly on the MR samples collected by the driving mode. When these MR samples are rather spare in terms of their locations and uneven timestamp intervals, the two sequence models are hard to capture spatiotemporal locality.

(2) Timestamp Intervals: In this experiment, we study the effect of timestamp intervals on localization errors. From the MR samples of the Jiading 2G dataset, we randomly select some samples to make sure that the timestamp difference between neighbouring MR samples in MR sequences is smaller than a certain time interval. As shown in Figure 7, we vary the time interval from 3 seconds to 120 seconds and compare the PRNet against another two sequence models: SeqtoSeq and HMM. From this figure, we have the following result. Firstly, a higher time interval (and thus more sparse data samples) leads to the higher prediction errors of all three sequence models. Secondly, the growth trend of PRNet median errors is rather smooth. It is mainly because PRNet can capture temporal dependencies even from sparse samples. In addition, HMM suffers from much higher errors than SeqtoSeq, indicating that the deep sequence models including SeqtoSeq and PRNet are better to capture long-term dependencies than the first-order HMM model. Then, even with high timestamp intervals from 30 to 120 seconds, PRNet can still perform well while the two other models not, mainly due to the time interval and subsequence attention mechanism in PRNet.

Recall that we might merge multiple neighbouring MR sequences to a single one such that each (merged) MR sequence contains at least τ samples (see Section 3.1). To this end, Figure 7 plots the number of merged MR sequences. When the timestamp interval varies from 3 to 120 seconds, the number of merged MR sequences grows from 2.2 to 4.15 on average. Meanwhile, each (merged) MR sequence contains at least τ MR samples from 12 to 5 (due to space limit, we do not plot the number τ in this figure). Since the timestamp intervals vary from 1 to 125 seconds in our datasets, we require that each MR sequence contains at least $\tau = 5$ MR samples. In this way, the MR samples in each (merged) sequence exhibit spatiotemporal locality even with high timestamp intervals. In addition, we are interested in the performance of PRNet against the one without merging MR sequences (denoted by PRNet$^-$). We find that PRNet$^-$ suffers from slightly higher errors, e.g., 16.45% higher median error for the timestamp interval 120 seconds. As a summary, merging MR sequences can capture spatiotemporal locality from sparse MR samples and improve the localization errors of PRNet.

(3) Base Station Density: We now study how the density of base stations affects the prediction errors of PRNet by varying the percentage of base stations from 25%-100%. Specifically, among all base stations in a MR dataset, we randomly remove some base stations from MR samples. If a certain MR sample contains a removed base station, then the signal measurements regarding this base station are dropped from the sample. Figure 8 indicates that PRNet leads to a competitive accuracy even if we drop 50% base stations. Nevertheless, a lower base station density means higher errors. The reason is that sparse base stations incur spatial ambiguity of MR samples. More base stations in MR samples alternatively lead to more discriminative MR features and lower errors.

(4) Ablation Study of PRNet: To study the importance of two predictors in PRNet, we design the following variants of PRNet. 1) PRNet$_l$ using the local predictor alone (which can be treated as a single-point-based predictor to process MR samples individually): we need to add an additional dense layer to produce the final outputs; 2) PRNet$_g$ using the global predictor alone: we need to reshape the original input matrix $\boldsymbol{X}_{i,j} \in \mathbb{R}^{F \times N}$ of a MR sequence into a column vector $(F \times N) \times 1$, which is directly fed into the global predictor. As shown in Figure 9, PRNet$_l$ incurs a high growth of localization errors, indicating the significant contribution offered by the global predictor to capture the temporal dependencies of the entire sequence. Instead, PRNet$_g$ could reduce the top 90% errors, when compared to PRNet$_l$. It mainly because that the dependencies among neighbouring MR samples can mitigate the prediction of outliers. Not surprisingly,

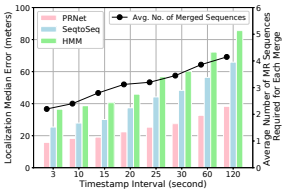


Figure 7: Timestamp Interval

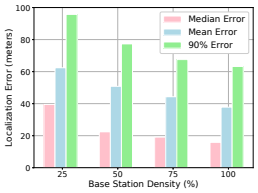


Figure 8: Base Station Density

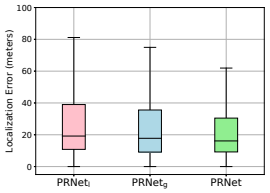


Figure 9: PRNet Components

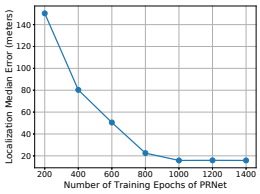


Figure 10: Training Epochs

PRNet with the local and global predictors can lead to the best result.

(5) Number of Training Epochs: Figure 10 gives the effect of used training epochs in PRNet. From this figure, we find that more epochs from 200 to 800 lead to significant reduction of localization errors on test data. Nevertheless, the curve remains stable when training epochs are more than 1000. Thus, 1000 training epochs could be enough to achieve acceptable results.

(6) Learning Rate: Table 6 shows the effect of different learning rates on PRNet. Learning rate determines whether the objective function can converge to a local minimum and how fast it converges. A carefully tuned learning rate enables the objective function to converge to a local minimum in a reasonable training epoches. From this table, we find that a fixed learning rate may not lead to the best accuracy. It is mainly because at the beginning of the training process, PRNet is far from the convergence, and a great learning rate is helpful; and after a certain number of epochs, PRNet is close to the convergence and a small learning rate is beneficial. Thus, following the previous work [24], we adopt a step-decay technique to drop the current learning rate by a factor every a certain number of epochs. The experimental result indicates that the step decay-based learning rate leads to more reliable result and faster convergence (a smaller number of training epoches are needed) than the scheme with a fixed learning rate.

Learning Rate	0.0001	0.0005	0.001	step-decay
Median Error (meters)	16.3	16.6	19.8	15.8
Mean Error (meters)	39.0	38.9	42.3	37.8
90% Error (meters)	65.3	65.7	66.7	63.2
# of Training Epochs	2500	1800	1300	1000

Table 6: Effect of Learning Rate

5.4 Visualization

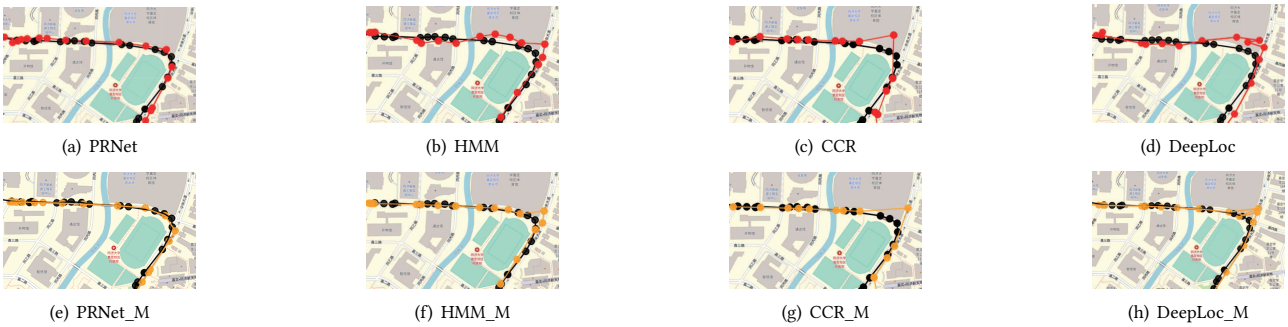
Finally, Figure 11 visualizes the prediction results of PRNet and three typical Telco location recovery approaches (HMM, CCR, and DeepLoc) on a randomly selected small area from the Jiading 2G data set. As shown in this figure, the trajectory predicted by PRNet is the closest to the ground truth in terms of horizontal moving directions and vertical shift out of road segments. Yet DeepLoc leads to the most shift in both horizontal and vertical directions. The median localization errors of HMM, CCR, and DeepLoc are very close, but HMM leads to a smoother trajectory of visualization results. It is mainly because HMM first utilized Hidden Markov Model for extracting the contextual dependencies between neighboring MR samples and next the filtering technique for post-processing.

In addition, with help of the extra road network, we perform the map-matching technique [28] on the trajectories and plot the map-matching trajectories in this figure. This post-processing technique can greatly improve CCR and DeepLoc to generate smooth trajectories. However, the map-matching technique cannot fully overcome the overall shift along the horizontal moving direction, which has been clearly visualized in this figure. Consistent with the result in Section 5.2, PRNet leads to acceptable results, even without those post-processing steps such as map-matching.

6 RELATED WORK

Trajectory Recovery: Unlike single point-based localization, the sequence localization of (GPS) position trajectories, namely trajectory recovery problem, frequently exhibit much lower localization errors. For example, AT&T researchers [15] proposed an outdoor localization approach, which first exploits the hidden markov model (HMM) for trajectory recovery and then applies particle filters to generate smooth trajectories. CCR [29] designed temporal and spatial contextual features (e.g., moving speeds, time gap, etc.) based on domain expert knowledge and feature engineering expertise to greatly improve prediction accuracy. In addition, the previous work [21] proposed a regularization framework to reconstruct mobile trajectories from sparse location fingerprints by enforcing two properties including spatial and temporal smoothness. Similar to [21], map-matching techniques and their variants [28] expect to recover an entire trajectory from a set of (sparse) locations, e.g., GPS coordinates. These works, with help of road network constraints, project the locations onto road networks to correct outlier locations (for example noisy GPS points) and recover an entire trajectory. Our work differs from [21, 21] which assume the availability of sparse location points within the to-be-recovered trajectories.

Deep Learning for Localization Systems: Recently, deep neural networks have been used in indoor or outdoor localization systems. Firstly, in dynamic indoor environment, the Deep Belief Network (DBN) and Gaussian-Bernoulli Restricted Boltzmann Machines have been utilized in fingerprinting-based indoor localization to increase estimation accuracy and reduce generalization error [5]. In addition, to process the RSS time-series acquired from wireless local area network (WLAN) access points, the previous work [8] leveraged a convolutional neural network (CNN) which is fed with RSS feature matrices for indoor localization by extracting the temporal dependencies between the last RSS readings. DeepFi [23] is another deep-learning-based indoor fingerprinting system based on channel state information (CSI) rather than RSS for indoor localization. DeepFi explores the features of wireless channel data and obtains the optimal weights of DNN as fingerprints which effectively describe the characteristics of CSI for each location and help reduce noise. These fingerprint-based approaches typically require sufficient



(a) PRNet (b) HMM (c) CCR (d) DeepLoc

(e) PRNet_M (f) HMM_M (g) CCR_M (h) DeepLoc_M

Figure 11: Visualization (Map Scale 1.5:10000). Black: ground truth; Red: predicted location; Orange: Map-matching location.

samples to construct fingerprint database, and do not work well for our problem with heterogenous samples. Finally, for outdoor Telco localization, DeepLoc [18] is the first to utilize deep learning techniques to estimate locations. In DeepLoc, the input vector contains the RSSI signal strength of all base stations within the entire area of interest, and the data augmentation technique is utilized to tackle the challenges of noisy data and insufficient training samples. Due to a large number of base stations, the input vector with very high dimensionality is rather sparse. In summary, all of these DNN-based works typically perform single-point-based localization in spite of the contextual or spatiotemporal features have been utilized [8] and in stead we focus on the sequence-based DNN model.

7 CONCLUSION

In this paper, we proposed a framework called PRNet to recover a smooth trajectory connecting precisely predicted outdoor locations. PRNet ensembles the power of CNN, LSTM, and two attention mechanisms to properly leverage all of the local, short- and long-term spatial and temporal dependencies from input MR sequences. Our extensive evaluation shows that PRNet greatly outperforms four state-of-the-art approaches and alternative variants of PRNet on six datasets collected in three representative areas in Shanghai. The promising result of PRNet inspires future studies on issues such as how to use the recovered trajectories from Telco MR data for mobility analysis and the potentials of applying PRNet to forthcoming 5G networks.

Acknowledge: This work is partially supported by National Natural Science Foundation of China (Grant No. 61572365, No. 61772371, No. 61672380), and The Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Google maps for mobile. http://www.google.com/mobile/maps.
[2] R. A. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, and C. Volinsky. Human mobility characterization from cellular network data. *Commun. ACM*, 56(1):74–82, 2013.
[3] R. A. Becker, R. Cáceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Computing*, 10(4):18–26, 2011.
[4] C. Costa and D. Zeinalipour-Yazti. Telco big data: Current state & future directions. In *IEEE MDM 2018*, pages 11–14, 2018.
[5] G. Felix, M. Siller, and E. Navarro-Alvarez. A fingerprinting indoor localization algorithm based deep learning. In *ICUFN 2016*, pages 1006–1011. IEEE, 2016.
[6] J. Hannink, T. Kautz, C. F. Pasluosta, J. Barth, S. Schülein, K. Gaßmann, J. Klucken, and B. M. Eskofier. Mobile stride length estimation with deep convolutional neural networks. *IEEE J. Biomedical and Health Informatics*, 22(2):354–362, 2018.
[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
[8] M. Ibrahim, M. Torki, and M. ElNainay. CNN based indoor localization using RSS time-series. In *2018 IEEE Symposium on Computers and Communications, ISCC 2018, Natal, Brazil, June 25-28, 2018*, pages 1044–1049. IEEE, 2018.
[9] M. Ibrahim and M. Youssef. A hidden markov model for localization using low-end GSM cell phones. In *IEEE ICC 2011*, pages 1–5. IEEE, 2011.
[10] M. Ibrahim and M. Youssef. Cellsense: An accurate energy-efficient GSM positioning system. *IEEE Trans. Vehicular Technology*, 61(1):286–296, 2012.
[11] lames J. Caffery and G. L. Stuber. Overview of radiolocation in cdma cellular systems. *IEEE Communications Magazine*, 36(4):38–45, Apr. 1998.
[12] I. Leontiadis, A. Lima, H. Kwak, R. Stanojevic, D. Wetherall, and K. Papagiannaki. From cells to streets: Estimating mobile paths with cellular-side data. In *CoNEXT*, pages 121–132. ACM, 2014.
[13] R. Margolies, R. A. Becker, S. D. Byers, S. Deb, R. Jana, S. Urbanek, and C. Volinsky. Can you find me now? evaluation of network-based localization in a 4g LTE network. In *IEEE INFOCOM 2017*, pages 1–9, 2017.
[14] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.
[15] A. Ray, S. Deb, and P. Monogioudis. Localization of LTE measurement records with missing information. In *IEEE INFOCOM 2016*, pages 1–9, 2016.
[16] G. J. Scott, R. A. Marcum, C. H. Davis, and T. W. Nivin. Fusion of deep convolutional neural networks for land cover classification of high-resolution imagery. *IEEE Geosci. Remote Sensing Lett.*, 14(9):1638–1642, 2017.
[17] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810, 2015.
[18] A. Shokry, M. Torki, and M. Youssef. Deeploc: a ubiquitous accurate and low-overhead outdoor cellular localization system. In *SIGSPATIAL/GIS*, pages 339–348. ACM, 2018.
[19] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS 2014*, pages 3104–3112, 2014.
[20] S. Swales, J. Maloney, and J. Stevenson. Locating mobile phones and the us wireless e-911 mandate. In *Novel Methods of Location and Tracking of Cellular Mobiles and Their System Applications (Ref. No. 1999/046), IEE Colloquium on*, pages 2–1. IET, 1999.
[21] D. A. Tran, T. Zhang, and S. Gong. A regularization framework for fingerprint-based reconstruction of mobile trajectories. *IJPEDS*, 31(3):268–279, 2016.
[22] R. M. Vaghefi, M. R. Gholami, and E. G. Ström. Rss-based sensor localization with unknown transmit power. In *IEEE ICASSP 2011*, pages 2480–2483.
[23] X. Wang, L. Gao, S. Mao, and S. Pandey. Csi-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Vehicular Technology*, 66(1):763–776, 2017.
[24] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. NIPS 2017. pages 4151–4161, 2017.
[25] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy. Hierarchical attention networks for document classification. In *HLT-NAACL*, pages 1480–1489. The Association for Computational Linguistics, 2016.
[26] Z. Yuan, X. Zhou, and T. Yang. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In Y. Guo and F. Farooq, editors, *ACM SIGKDD 2018*, pages 984–992.
[27] Y. Zhang, W. Rao, and Y. Xiao. Deep neural network-based telco outdoor localization. In *ACM SenSys 2018, Shenzhen, China, November 4-7, 2018*, pages 307–308, 2018.
[28] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban computing: Concepts, methodologies, and applications. *ACM TIST*, 5(3):38:1–38:55, 2014.
[29] F. Zhu, C. Luo, M. Yuan, Y. Zhu, Z. Zhang, T. Gu, K. Deng, W. Rao, and J. Zeng. City-scale localization with telco big data. In *ACM CIKM 2016*, pages 439–448.