The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

DeepUrbanMomentum: An Online Deep-Learning System for Short-Term Urban Mobility Prediction

Renhe Jiang,[†] Xuan Song,[†‡*] Zipei Fan,[†] Tianqi Xia,[†] Quanjun Chen[†]
Satoshi Miyazawa,[†] Ryosuke Shibasaki [†]
[†]Center for Spatial Information Science, The University of Tokyo
[‡]Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology
{jiangrh, songxuan, xiatianqi, koitaroh, shiba}@csis.u-tokyo.ac.jp, {fanzipei, chen1990}@iis.u-tokyo.ac.jp

Abstract

Big human mobility data are being continuously generated through a variety of sources, some of which can be treated and used as streaming data for understanding and predicting urban dynamics. With such streaming mobility data, the online prediction of short-term human mobility at the city level can be of great significance for transportation scheduling, urban regulation, and emergency management. In particular, when big rare events or disasters happen, such as large earthquakes or severe traffic accidents, people change their behaviors from their routine activities. This means people's movements will almost be uncorrelated with their past movements. Therefore, in this study, we build an online system called DeepUrbanMomentum to conduct the next short-term mobility predictions by using (the limited steps of) currently observed human mobility data. A deep-learning architecture built with recurrent neural networks is designed to effectively model these highly complex sequential data for a huge urban area. Experimental results demonstrate the superior performance of our proposed model as compared to the existing approaches. Lastly, we apply our system to a real emergency scenario and demonstrate that our system is applicable in the real world.

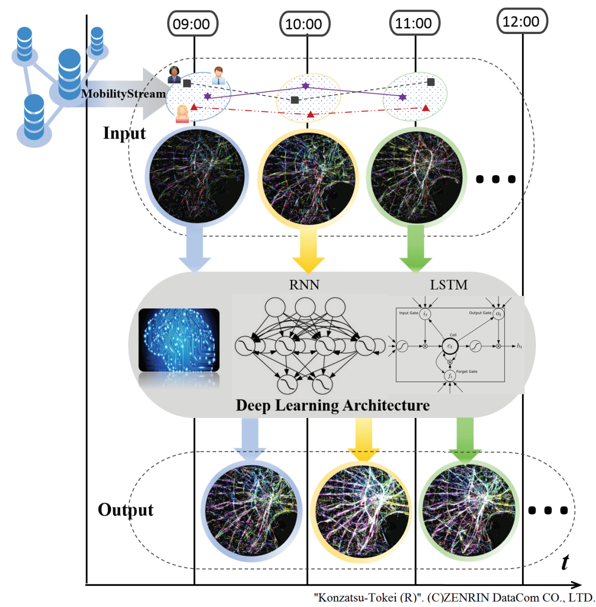


Figure 1: Can we develop an online intelligent system for short-term human mobility prediction with high precision by using recent momentary mobility at a citywide level? Big human mobility data and deep-learning technologies provide us with the opportunity to implement this system.[1]

Introduction

The next-generation 5G mobile Internet technologies will mark a new era in the information industry, and they will play an important role in stimulating the growth of the Internet of Things (IoT). Against this background, massive GPS trajectories that are being continuously generated from sources, such as smartphones, GPS devices on cars, WLAN networks, and location-based social networks, become important for use as real-time human mobility data streams. With such valuable streaming data, people's future behaviors and movements can be predicted step-by-step in an online manner, based on an intuitive Markov-like assumption that people's next behaviors mostly rely on their recent ones. Especially, when big rare events or disasters, such as high-magnitude earthquakes happen, people's behaviors and movements will become rather different from their daily routines. Such online short-term predictions using recent momentary mobility will become very necessary and practical. Elevating this to a citywide level, namely predicting *UrbanMomentary* human mobility for a huge urban area, can play a crucial role in effective urban planning, transportation scheduling, and emergency management.

However, even for a short period, human mobility and transportation transitions for a large-scale transportation system are highly complex, which are almost impossible to be effectively modeled using classical methodologies or simple neural network-based models. Emerging deep-learning technologies have demonstrated superior performances on various datasets (e.g., images, texts, and videos) (Bengio 2009; Vincent et al. 2010; Demuth et al. 2014) of existing classical approaches. Hence, in this study, we investigate the various aspects of human mobility during a short period in a large urban area by using a deep-learning-based

*Corresponding author

approach. We also develop an intelligent system for citywide short-term human mobility prediction with high precision compared with the existing approaches.

In this study, we first collect big human mobility data and process them into calibrated trajectories, and construct an artificial human mobility data stream for a large urban area. Then, we build an online intelligent system called *DeepUrbanMomentum* to continuously take the recent momentary mobility as the input and predict next short-term urban human mobility as the output, as shown in Fig.1. The modeling component of our system is based on the deep Recurrent Neural Network (RNN) architecture constructed using two layers: one RNN layer is used to turn the inputted location sequence into a single latent vector containing information about the entire sequence. Then, a functional layer will repeat this latent vector multiple times and pass this vector sequence to another RNN layer that is used to turn this constant sequence into multiple steps of output mobility. This deep model is built essentially as a regression model that can directly take continuous values (location coordinates) as input and output. Finally, given an artificial mobility data stream for a big urban area, DeepUrbanMomentum will automatically conduct an online deep-learning process and report the prediction results of UrbanMomentum by a well-trained model by using the current urban mobility data. To the best of our knowledge, DeepUrbanMomentum is the first system that applies the deep-learning approach to effectively perform online short-term human mobility predictions at a citywide level. It has the following key characteristics:

- It is built and tested based on a big human mobility data source, which stores the GPS records of 1.6 million users over three years.
- It is built as an online prediction system driven by mobility stream and deep-learning technologies.
- It constructs a deep-sequence learning model with RNN for effective multi-step predictions.
- It is applied to real-world scenarios and verified as a highly deployable prototype system.

The remainder of this paper is organized as follows: Section 2 gives an overview of our data source. Section 3 gives the definition of UrbanMomentum and the prediction model. Section 4 explains the modeling details and the deep-learning architectures. Section 5 shows the experimental details, the performance evaluation, and the prediction results in a real-world scenario. Section 6 introduces studies related to our research. Section 7 contains summaries, the limitations of our current system, and our future work.

Data Source

A raw GPS log dataset was collected anonymously from approximately 1.6 million mobile phone users in Japan over a three-year period (August 1, 2010 to July 31, 2013)[1]. Data collection was conducted by a mobile operator and a private company under an agreement with the mobile phone users. This dataset contains approximately 30 billion GPS records, and the total size of the data is more than 1.5 TB. To better simulate a real-time situation for our online system, this dataset is stored on a Hadoop cluster, containing 32 cores, 32 GB memory, and 16 TB storage, which can run 28 tasks simultaneously. Furthermore, we use Hive on top of Hadoop to make the whole system support SQL-like spatial queries. Therefore, GPS trajectories of a specified city and day can be retrieved in a short response time, and our database can be regarded as a nearly real-time data source that can provide streaming trajectory data to our online system.

Preliminaries

Definition 1 (Raw human trajectory): The raw trajectory collected from an individual is essentially a sequence of 3-tuple: $(timestamp, latitude, longitude)$, which can indicate a person's location according to a captured timestamp. In the rest of this paper, it is further simplified as a sequence of (t, l)-pairs.

Note that the raw trajectory has a lot of temporal uncertainties because of different time intervals between two consecutive timestamps. Our goal is to predict citywide human movements; therefore, it motivates us to reduce temporal uncertainty by calibrating the raw trajectory to have equal time intervals Δt, which is defined as follows:

Definition 2 (Calibrated human trajectory): A calibrated human trajectory $traj$ from time t_1 to t_m is a sequence of timestamp-location pairs denoted as: (t_1, l_1), (t_2, l_2), ..., (t_m, l_m) that satisfies:

$$\forall i \in [1, m), |t_{i+1} - t_i| = \Delta t$$

In fact, this calibration operation is performed based on the following assumption.

Definition 3 (Temporal certainty assumption): For each individual person, his/her location coordinates can be retrieved every Δt time.

Definition 4 (Urban human mobility stream): Based on the above definitions, urban human mobility can be regarded as a kind of streaming data arriving every Δt time interval, from which we can get n infinite human trajectories corresponding to n individual persons. Furthermore, these n trajectories will all be spatially contained in one urban area denoted as ur. Therefore, an urban human mobility stream is determined by three parameters ur, n, and Δt, which are given as:

$$uhms = F(ur, n, \Delta t)$$

Ideally, n should be the total number of a city's resident population, and Δt should be several seconds. However, this ideal mobility stream is extremely hard to build because of various limitations of location acquisition technologies. Therefore, n is more likely to be the total number of active

[1]"Konzatsu-Tokei (R)" from ZENRIN DataCom CO.LTD is used by us, which refers to people flows data collected by individual location data sent from mobile phone with enabled AUTO-GPS function under users' consent, through the "docomo map navi" service provided by NTT DOCOMO, INC. Those data is processed collectively and statistically in order to conceal the private information. Original location data is GPS data (latitude, longitude) sent in about every a minimum period of 5 minutes and does not include the information to specify individual such as gender or age. In this study, the proposed methodology is applied to raw GPS data by NTT DOCOMO, INC.

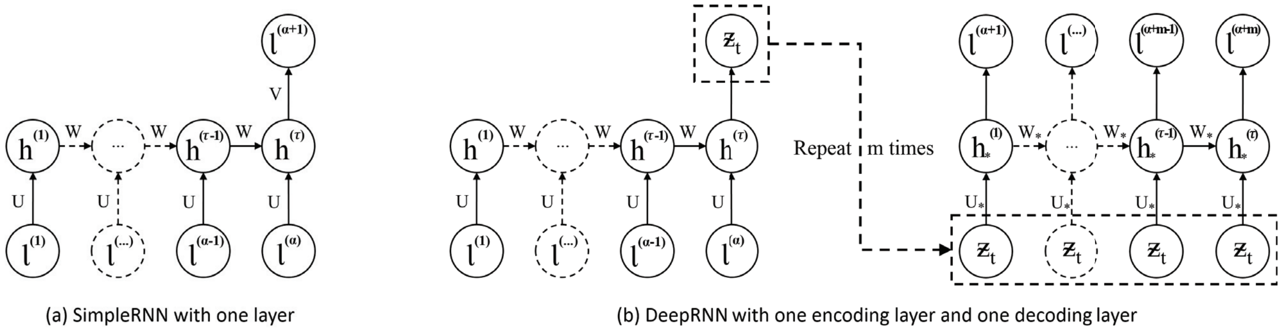


Figure 2: Deep Sequential Modeling Architecture.

users of a certain smartphone application, and Δt is set to 10 min or a longer time interval.

Definition 5 (Current urban mobility): given an $uhms$, a current time t and an integer α, current urban mobility X_t is defined as follows:

$$X_t = \{traj \mid traj \in uhms \wedge \forall i, t - \alpha\Delta t \leq traj.t_i \leq t\}$$

which intuitively means current α steps of urban human mobility accumulated from $uhms$.

Definition 6 (Next urban mobility): Similarly, given an $uhms$, a current time t, and an integer β, the next urban mobility X_{t+1} is defined as follows:

$$X_{t+1} = \{traj \mid traj \in uhms \wedge \forall i, t + \Delta t \leq traj.t_i \leq t + \beta\Delta t\}$$

which means the next β steps of urban human mobility.

Definition 7 (UrbanMomentum prediction model): Given the current urban mobility X_t, UrbanMomentum prediction will construct a model $P_\theta(\widehat{X}_{t+1} \mid X_t)$, in which θ represents a set of model parameters and $\widehat{X}_{t+1}$ is the predicted next urban mobility. It will be built as a regression model, and its parameters can be obtained by minimizing the prediction error $L(\widehat{X}_{t+1}, X_{t+1})$ as follows:

$$\theta = \underset{\theta}{\operatorname{argmin}}\, L(\widehat{X}_{t+1}, X_{t+1}) = \underset{\theta}{\operatorname{argmin}}\, ||\widehat{X}_{t+1} - X_{t+1}||^2$$

Short-Term Urban Mobility Modeling

Our system is built based on a given $uhms$ to predict the next urban mobility $\widehat{X}_{t+1}$ using X_t in an online manner. It is a relatively easy task when $\widehat{X}_{t+1}.\beta = 1$, which means the online system accumulates the current α steps of the urban human mobility X_t and uses them to predict only one-step of the next urban mobility. However, it is always not sufficient to give out only the one-step prediction, especially during times of emergencies, such as rare events or some natural disasters. Therefore, a more meaningful prediction $\widehat{X}_{t+1}$ with a large β called Short-Term Urban Mobility Prediction becomes the main task of our online prediction system.

SimpleRNN Modeling Architecture

Given a X_t and based on our temporal certainty assumption, a current mobility of one person can be simplified as: $x_t = l_1, l_2, ..., l_\alpha$, and similarly a next short-term prediction can be represented as: $\widehat{x}_{t+1} = l_{\alpha+1}, l_{\alpha+2}, ..., l_{\alpha+\beta}$. It can be further modeled as:

$$P(l_{\alpha+1}, l_{\alpha+2}, ..., l_{\alpha+\beta}) = \prod_{i=1}^{\beta} P(l_{i+\alpha} \mid l_i, l_{i+1}, ..., l_{i+\alpha-1}) \tag{1}$$

Spatial Continuity. This model is similar to the n-gram model, which is a typical probabilistic sequential model for predicting the next item in such a sequence in the form of a (n-1)-order Markov model. However, the longitude and latitude of each location l is a continuous value in our problem definition because of the spatial continuity for which it is not simple to utilize the Markov model. Some may suggest a classical methodology that partitions the whole area into massive grids to convert the continuous space into discrete values. It is still difficult because our online system has to predict short-term human mobility for large urban areas, such as the Great Tokyo Area. Even with 1000-meter meshing, it still generates about 4,000 grids for the whole urban area (3,925 km^2), which will lead to an extremely sparse transition matrix if we apply the Markov model based on this huge mesh. In conclusion, urban human mobility on a continuous large-scale area is a highly complex phenomenon, which cannot be modeled without using classical methodologies. The above information motivates us to employ deep-learning technologies, such as RNNs (Demuth et al. 2014), and their special variants of the long short-term memory (LSTM) networks (Hochreiter and Schmidhuber 1997), to our system for mobility modeling. These have provided an impressive performance in modeling sequential data, such as speech and text. In particular, for our system, they can help us model human mobility on a continuous large urban space in a regression manner.

Recurrent Neural Networks. Compared with traditional neural networks, RNNs are specially designed for sequential data modeling. In traditional neural networks, neurons in one layer and its neighboring layers are fully connected, whereas neurons in the same layer do not have any connections. Such structures cannot effectively deal with the situation when data are not independent, such as words in a sentence. The typical structure of a simple RNN is shown in Fig.1. We can see that the neighboring neurons in the same hidden layer are connected with one another so that the network can memorize former information and have an impact on the output of the current timestep τ. Therefore, the total input not only contains the input at the timestep τ, but also the output at the timestep τ-1. To train an RNN, the standard method is "backpropagation through time" (BPTT).

Our goal is to build an Urban Mobility Model for short-term prediction described by Equation (1) using an RNN. A simple recurrent network structure is depicted in Fig.2-(a), which typically contains an input layer, a hidden layer and an output layer, where $tanh$ is used in the hidden layer for mapping inputs into a s single latent vector also called the latent representation. $ReLU$ is used as the final activation function, and mean-squared-error (mse) is the objective function defined in Definition 8.

The formulas that govern the whole computation in our architecture are as follows:

$$s_\tau = \tanh(Ui_\tau + Ws_{\tau-1}) \tag{2}$$

$$o_\tau = \mathrm{ReLU}(Vs_\tau) \tag{3}$$

where i_τ represents the input $(l_i, l_{i+1}, ..., l_{i+\alpha-1})$, o_τ represents the output $l_{\alpha+i}$ described in Equation (1), W and U are weight matrices in the hidden layer and V is weight matrix in the output layer. All these weight parameters will be determined by applying the BPTT algorithm as mentioned above; the algorithm details will be omitted in this paper.

With this construction, an n-gram-like mobility regression model called SimpleRNN is built; the model can take continuous value of location coordinates as input and output.

DeepRNN Modeling Architecture

Short-Term mobility can be modeled and computed as defined in (1) in an iterative one-by-one manner. One major limitation of this model is to predict a relatively long short-term mobility. With the iteration going on, the accumulated iteration error will become large, which can result in terrible performance on the last several predicted steps. To tackle this problem, we improve the multi-step-to-one-step modeling in (1) with multi-step-to-multi-step modeling aimed at achieving better performance on "long" short-term predictions. This is defined as follows:

$$\begin{aligned} & P(l_{\alpha+1}, ..., l_{\alpha+\beta}) \\ & = \prod_{i=0}^{\lceil \frac{\beta}{m} \rceil - 1} P(l_{\alpha+i\cdot m+1}, ..., l_{\alpha+i\cdot m+m} \mid l_{1+i\cdot m}, ..., l_{\alpha+i\cdot m}) \end{aligned} \tag{4}$$

where m is multiple output steps at one time.

To deliver this idea, a deep-learning architecture called DeepRNN is constructed as shown in Fig.2-(b). It works in the following steps: (1) the first hidden layer of RNN maps the α steps of the inputted mobility into a single latent vector h, which contains information about the entire sequence; (2) this vector is repeated m times; and (3) another hidden layer of RNN is used to turn this constant sequence into the m steps of the output mobility. Similarly, SimpleRNN, $tanh$, and $ReLU$ are used as activation functions in these two RNN layers. Our deep architecture is similar to a sequence-to-sequence model (Sutskever, Vinyals, and Le 2014), and the two RNN layers act as an encoder and a decoder.

Experiment

Experimental Setup and Parameter Setting

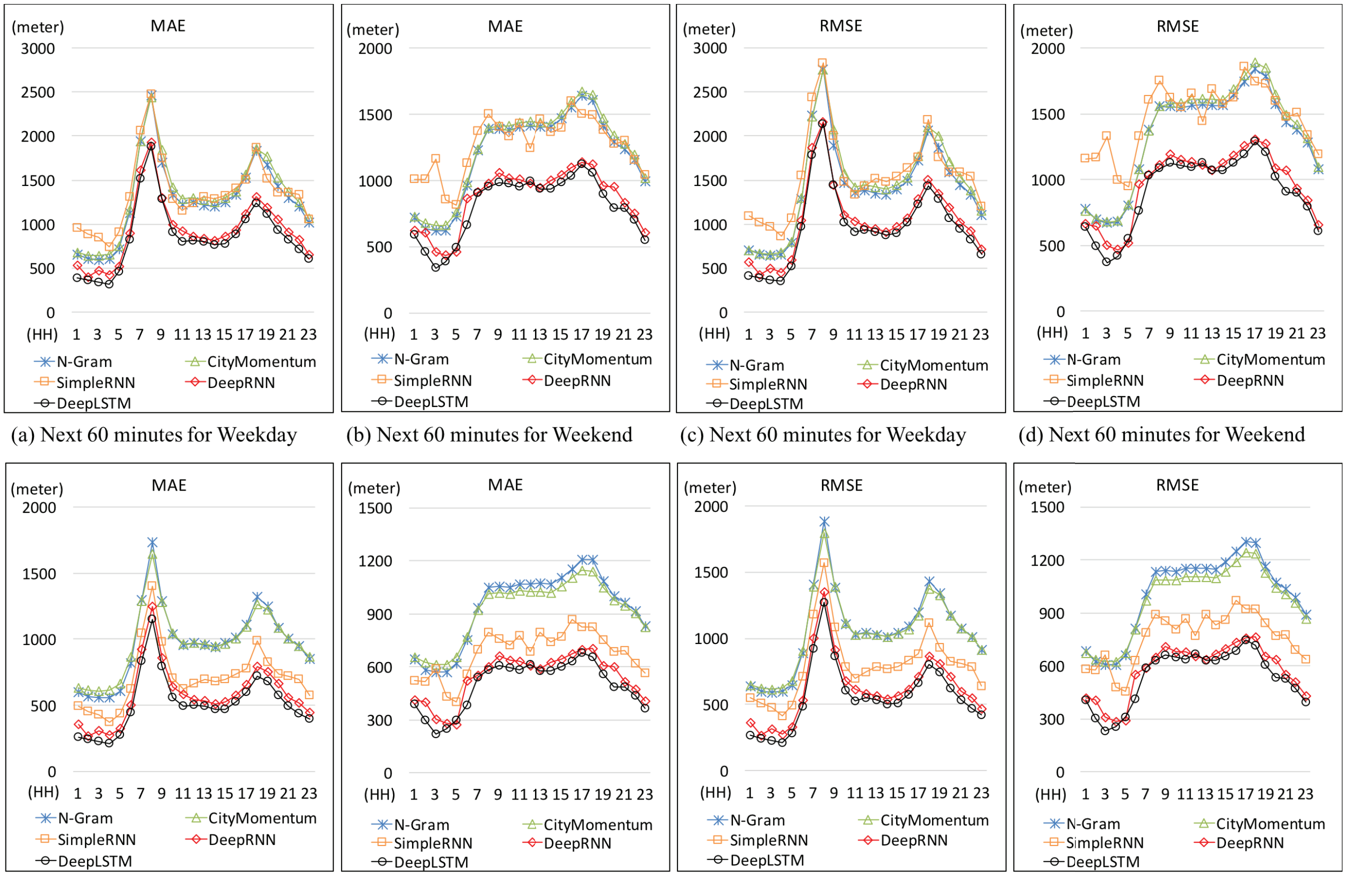
From our big human mobility database, we select one month of data (October 2011) and divide them into two parts, weekday dataset and weekend dataset, since urban human behaviors on weekdays and weekends are distinct from each other. Based on these, we construct two independent urban human mobility streams, denoted as $uhms_d$ (weekday) and $uhms_e$ (weekend), respectively, where $uhms_d.n \approx 112,360$ and $uhms_e.n \approx 94,812$ averaged by each day, $uhms_d.\Delta t$ and $uhms_e.\Delta t$ are both set to be 10 min, $uhms_d.ur$ and $uhms_e.ur$ are both set to the Greater Tokyo Area by default (Long. $\in$ [139.5, 139.9], Lat. $\in$ [35.5, 35.8]).

Then, the two types of UrbanMomentum predictions are tested on these two streams. One is called "Next 60 minutes" with $\widehat{X}_{t+1}.\beta$ equal to 6, and another is called "Next 30 minutes" with $\widehat{X}_{t+1}.\beta$ equal to 3. This means our system will predict the next-one-hour or next-half-hour urban human mobility in each report. Based on the empirical tuning result, we found the current urban mobility $X_t.\alpha$ = 3 and m = 3 in DeepRNN would be appropriate.

Lastly, all settings about modeling training and testing are kept the same in these two cases. For both SimpleRNN and DeepRNN, a 64-dimension vector is used as the latent representation Z_t of the entire X_t. The RMSprop algorithm is adopted in our system to govern the whole training process. We randomly select 80% of the data for model training and use the remaining 20% for validation, which is used to early-stop our training algorithm if the validation error is converged. This early-stopping strategy is very crucial for an online learning system like ours. Python and some Python libraries including Keras(Chollet 2015) and TensorFlow(Abadi et al. 2015) are used to implement our system.

Performance Evaluation

Comparison models: (1) N-Gram. It is a widely used algorithm for modeling sequential data, especially in the filed of natural language processing. In our study, we applied this model basing on a gridded space to predict next possible grid. Then the location coordinates were generated randomly inside the predicted grid from a uniform distribution. In order to avoid sparsity problem on a large urban area, we utilized Four-Gram model with $\Delta Long.$=0.01 $\times$ $\Delta Lat.$=0.008 (approximately $900m \times 900m$) as the mesh-size. (2) CityMomentum (Fan et al. 2015). It was firstly proposed for this kind of momentary mobility prediction at the citywide level. It is a predicting-by-clustering framework using a mixture of multiple random Markov chains. Each of them is an improved first-order markov model that considers not only the next-step probability from one subject's movements, but also the probability based on the cluster's movements, where the cluster is a bunch of subjects sharing similar movements with the subject. The parameter settings used in our experiment were kept same with the original



(a) Next 60 minutes for Weekday (b) Next 60 minutes for Weekend (c) Next 60 minutes for Weekday (d) Next 60 minutes for Weekend

(e) Next 30 minutes for Weekday (f) Next 30 minutes for Weekend (g) Next 30 minutes for Weekday (h) Next 30 minutes for Weekend

Figure 3: Performance Evaluation of Weekday and Weekend.

paper. (3)∼(4) SimpleRNN and DeepRNN. These are the two models proposed by us. (5) DeepLSTM. We also implement another comparison model with LSTM(Hochreiter and Schmidhuber 1997) called DeepLSTM, which shares the same architecture with DeepRNN except that ordinary neurons in traditional RNNs are replaced with special computation blocks namely LSTM. It has shown superior performance to traditional RNNs for long time-series modeling; therefore, we want to test if it can further improve the performance for our short-term prediction system.

Evaluation metrics: For n trajectories in a given $uhms$, the next β steps of locations will be predicted by every report of our online system. Therefore, to evaluate the overall accuracy of simulation results in a simpler way, we redefine two different metrics, the mean absolute error (MAE) and the root-mean-square error (RMSE), as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{\beta}\sum_{j=1}^{\beta}||l_{ij} - \widehat{l}_{ij}||\right]$$

$$\mathrm{RMSE} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{\beta}\sum_{j=1}^{\beta}||l_{ij} - \widehat{l}_{ij}||^2\right]^{\frac{1}{2}}$$

where $||l - \widehat{l}||$ means the Euclidean distance between the real location and the predicted one for each trajectory at each step, which will be measured in meters.

Performance comparison: Using the two metrics above, we compared the performances of the baseline models and our proposed deep-learning-based models for each hour by averaging each day's result. The evaluation results are summarized in Fig.3, and the results of the "Next 60 minutes" and "Next 30 minutes" are listed as Fig.3-(a)∼(d) and Fig.3-(e)∼(h), respectively. We can see that the DeepRNN model outperformed N-Gram, CityMomentum and SimpleRNN in each subfigure, and the advantage over SimpleRNN is more obvious for a relatively long-term "Next 60 minutes" prediction than a short "Next 30 minutes" one. Furthermore, DeepLSTM achieved better performances than DeepRNN, which demonstrates that the traditional RNN is not sufficient for modeling short momentary human movements. All the models performed relatively badly around 8:00 am and 6:00 pm of weekday-the typical morning and evening rush hours in Tokyo. Urban human mobility running on a highly complicated transportation system will change drastically during these hours, and UrbanMomentum becomes hard to predict by just using a few recent observations, which are a major limitation of our system for the normal weekday scenario.

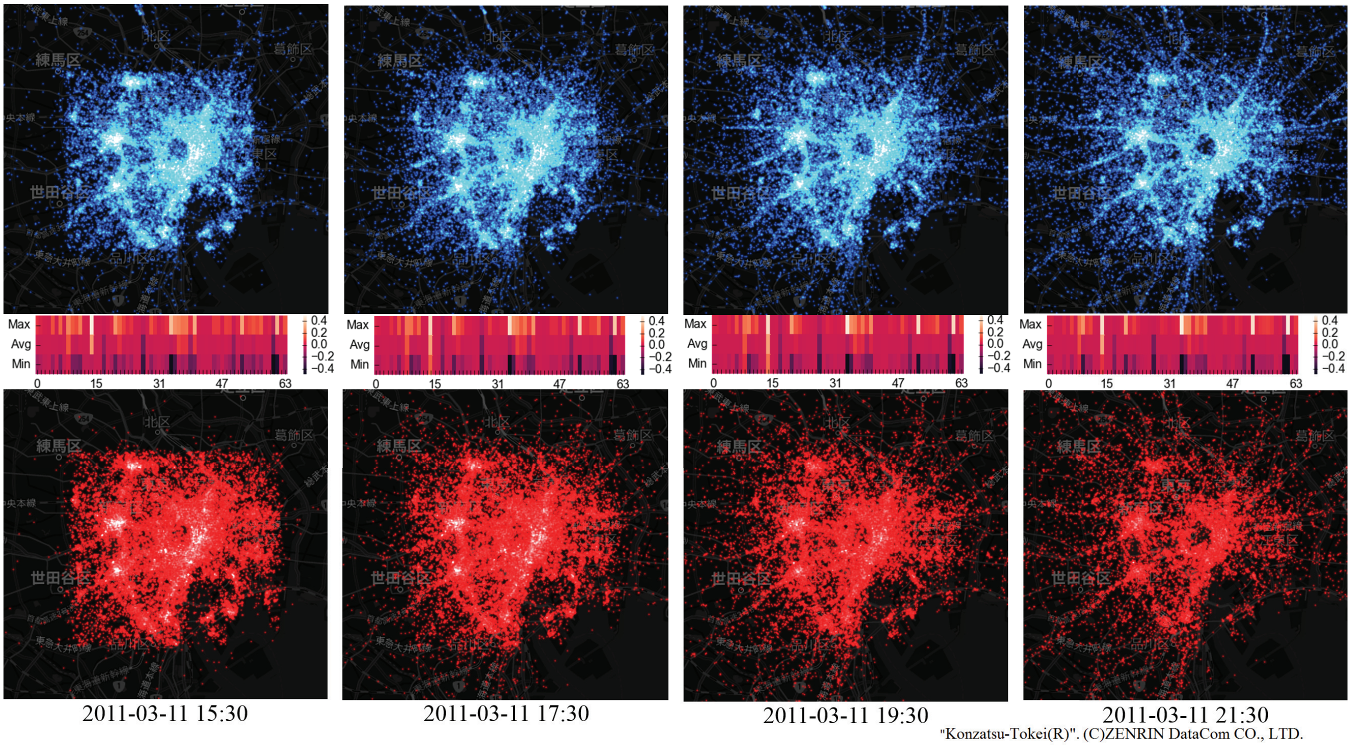


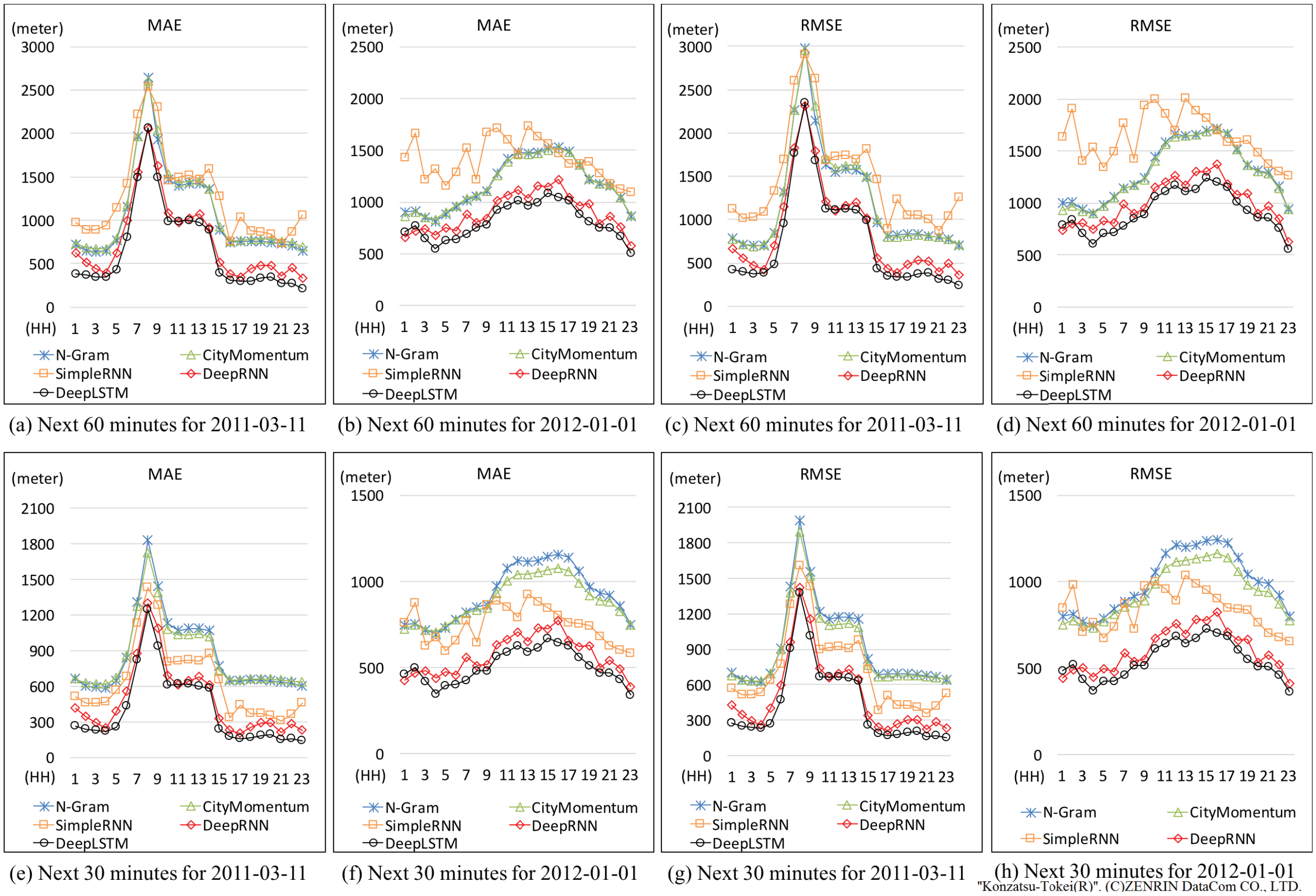
Figure 4: Visualization for human mobility in the core area of Tokyo in first six hours after the Great East Japan Earthquake. The prediction results are listed on the top in blue, and the corresponding ground truths are at the bottom in red. The 64-dimensional latent representations of UrbanMomentum learned by RNN at each timestamp are listed in the middle. The maximum, average and minimum are calculated across each dimension (0∼63) separately as a concise summary of the entire representations. [1]

Application to Real-World Scenario

We apply our prototype system to two real-world scenarios: **(1) 3.11 Japan Earthquake (2011-03-11)**. On March 11, 2011, at approximately 2:46 pm local time, the 9.0 magnitude Great East Japan Earthquake occurred off the east coast of Japan; this is considered one of the most powerful earthquakes worldwide. The earthquake caused a great impact on people's behaviors in the Great Tokyo Area. We apply our system to this major emergency scenario to validate its applicability by using a constructed mobility stream of 2011-03-11 00:00∼23:59. Our system gave out reports of "next 60 minutes" and "next 30 minutes" short-term urban mobility prediction every hour. Taking about 30,000 people who were in the core area of Tokyo at 3:00 pm as observation targets, we selected 15:30, 17:30, 19:30, and 21:30 (approximate 6 hours after the earthquake happened) as four evaluation timestamps, and compared their predicted locations with the ground truth. The visualization results are shown in Fig.4. This figure shows that our system can work with a relatively high accuracy level to predict urban human mobility after such a huge disaster and a slow evacuation process. Fig.4 also demonstrates the encoding RNN has effectively learned the latent representations of UrbanMomentum for the different timestamps after the earthquake. Furthermore, we used the same quantitative measures and summarized the evaluation results as Fig.5-(a)(c)(e)(g). Through the figures, we can see the different performance result around 3:00 pm comparing with normal weekdays because of the huge influence of the earthquake on urban transportation system. **(2) New Year's Day (2012-01-01)**. New Year can also be treated as a kind of rare event although it is not that rare as 3.11 earthquake. There are a number of New Year celebrations in Tokyo area, such as Disney Land New Year party and Shibuya square countdown. Especially, for "Hatsumode" (the first visit in Buddhist temple or shrine), a large crowd of people gather at Meiji Shrine, Sensoji Temple and Zojoji Temple, and most of the railway lines operate overnight on the New Year's Eve for this. All these make urban human behaviors very different from normal days. We constructed a mobility stream of 2012-01-01 00:00∼23:59 to test the performance of our system under this scenario and summarized the evaluation results as Fig.5-(b)(d)(f)(h). Different performance result during midnight (01:00∼05:00) can also be observed through the figures. DeepLSTM still achieved the best performances under both of the scenarios.

Related Work

CityMomentum (Fan et al. 2015) is the most closely related work with ours; however, our deep-learning-based approach can outperform it as shown in our experiments. Simulating human emergency mobility following disasters was addressed in (Song et al. 2014; 2015), but it required



(a) Next 60 minutes for 2011-03-11 (b) Next 60 minutes for 2012-01-01 (c) Next 60 minutes for 2011-03-11 (d) Next 60 minutes for 2012-01-01

(e) Next 30 minutes for 2011-03-11 (f) Next 30 minutes for 2012-01-01 (g) Next 30 minutes for 2011-03-11 (h) Next 30 minutes for 2012-01-01

Figure 5: Performance Evaluation of 3.11 Japan Earthquake and New Year' Day.

disaster information such as intensity of earthquake and damage level as additional input data. Modeling human mobility for very large populations (Song et al. 2010a; Isaacman et al. 2012; Konishi et al. 2016) are research topics close to ours, but still different from our problem definition. Moreover, traffic flow has also been studied in (Castro, Zhang, and Li 2012; Chen, Chen, and Qian 2014). However, all of these approaches did not use the power of deep learning technologies. Forecasting citywide crowd density (Hoang, Zheng, and Singh 2016; Zhang, Zheng, and Qi 2017) is a related endeavor based on deep learning, which builds a long time-series model for each region of a city, whereas our system predicts citywide short-term mobility based on the recent observations rather than a long-period's. Some researchers also have applied deep learning to predict traffic flow, traffic speed, congestion, and transportation mode along with human mobility (Huang et al. 2014; Lv et al. 2015; Ma et al. 2015a; 2015b; Song, Kanasugi, and Shibasaki 2016). Moreover, various studies conducted on human mobility data, are summarized as urban computing in (Zheng et al. 2014). C. Song (Song et al. 2010b) explored the upper bound of predictability of human mobility. J. Zheng (Zheng and Ni 2012) proposed an unsupervised learning algorithm for location prediction.

Conclusions

In this study, we collected big human mobility data to construct artificial mobility data streams for large urban area. Based on this, we build an intelligent system call DeepUrbanMomentum for online short-term mobility prediction at a citywide level based on momentary human movements that we achieved. A deep RNN was specially designed as an effective multistep-to-multistep prediction model. Experimental results demonstrated the superior performance of our proposed model compared to the existing approaches and other shallow models. Furthermore, we applied our system to a real-world scenario and verified its applicability.

Our system has some room for improvement in the following areas: (1) Our system is still struggling to deal with the situation when urban mobility is full of sudden changes such as morning rush hours. (2) Other heterogeneous data, such as transportation network and Point-of-Interest data, can also be used as auxiliary features for deep-learning models. (3) More sophisticated preprocessing will be included to improve the overall performance of our system. Particularly, we will apply map matching algorithm and trajectory calibration algorithm (Su et al. 2013) to improve the quality of the raw trajectories.

Acknowledgments

This work was partially supported by JST, Strategic International Collaborative Research Program (SICORP); Leading Initiative for Excellent Young Researchers (LEADER) Program and Grant in-Aid for Scientific Research B (17H01784) of Japans Ministry of Education, Culture, Sports, Science, and Technology (MEXT).

References

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

Bengio, Y. 2009. Learning deep architectures for ai. *Foundations and trends® in Machine Learning* 2(1):1–127.

Castro, P. S.; Zhang, D.; and Li, S. 2012. Urban traffic modelling and prediction using large scale taxi gps traces. In *Pervasive Computing*. Springer. 57–72.

Chen, P.-T.; Chen, F.; and Qian, Z. 2014. Road traffic congestion monitoring in social media with hinge-loss markov random fields. In *2014 IEEE International Conference on Data Mining*, 80–89. IEEE.

Chollet, F. 2015. keras. https://github.com/fchollet/keras.

Demuth, H. B.; Beale, M. H.; De Jess, O.; and Hagan, M. T. 2014. *Neural network design*. Martin Hagan.

Fan, Z.; Song, X.; Shibasaki, R.; and Adachi, R. 2015. Citymomentum: an online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 559–569. ACM.

Hoang, M. X.; Zheng, Y.; and Singh, A. K. 2016. Forecasting citywide crowd flows based on big data. *ACM SIGSPATIAL*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Huang, W.; Song, G.; Hong, H.; and Xie, K. 2014. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *Intelligent Transportation Systems, IEEE Transactions on* 15(5):2191–2201.

Isaacman, S.; Becker, R.; Cáceres, R.; Martonosi, M.; Rowland, J.; Varshavsky, A.; and Willinger, W. 2012. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 239–252. Acm.

Konishi, T.; Maruyama, M.; Tsubouchi, K.; and Shimosaka, M. 2016. Cityprophet: city-scale irregularity prediction using transit app logs. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 752–757. ACM.

Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; and Wang, F.-Y. 2015. Traffic flow prediction with big data: a deep learning approach. *Intelligent Transportation Systems, IEEE Transactions on* 16(2):865–873.

Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; and Wang, Y. 2015a. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54:187–197.

Ma, X.; Yu, H.; Wang, Y.; and Wang, Y. 2015b. Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one* 10(3):e0119044.

Song, C.; Koren, T.; Wang, P.; and Barabási, A.-L. 2010a. Modelling the scaling properties of human mobility. *Nature Physics* 6(10):818–823.

Song, C.; Qu, Z.; Blumm, N.; and Barabási, A.-L. 2010b. Limits of predictability in human mobility. *Science* 327(5968):1018–1021.

Song, X.; Zhang, Q.; Sekimoto, Y.; and Shibasaki, R. 2014. Intelligent system for urban emergency management during large-scale disaster. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Song, X.; Zhang, Q.; Sekimoto, Y.; Shibasaki, R.; Yuan, N. J.; and Xie, X. 2015. A simulator of human emergency mobility following disasters: Knowledge transfer from big disaster data. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Song, X.; Kanasugi, H.; and Shibasaki, R. 2016. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. IJCAI.

Su, H.; Zheng, K.; Wang, H.; Huang, J.; and Zhou, X. 2013. Calibrating trajectory data for similarity-based analysis. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, 833–844. ACM.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11(Dec):3371–3408.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. *Thirty-First AAAI Conference on Artificial Intelligence*.

Zheng, J., and Ni, L. M. 2012. An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 153–162. ACM.

Zheng, Y.; Capra, L.; Wolfson, O.; and Yang, H. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5(3):38.