Applying Deep Recurrent Neural Network to Predict Vehicle Mobility

Wei Liu†, and Yozo Shoji†
†Social-ICT System Laboratory, National Institute of Information and Communications Technology, Tokyo, Japan
Email: wei_liu, shoji@nict.go.jp

***Abstract*—Sensing data gathering and dissemination is one of the most challenging tasks in smart city construction, and vehicles moving around a city have been widely considered as a good candidate to deliver data efficiently and economically. Hence, this paper proposes a deep recurrent neural network-based algorithm to predict vehicle mobility and facilitate vehicle-based sensing data delivery. Extensive evaluations have been conducted by using a large-scale taxi mobility dataset that is obtained from a smart city testbed deployed in Tokyo, Japan. The results have validated that, compared with the most state-of-art algorithms, our proposal can improve the F1-Score of vehicle mobility prediction by a range of 18.3% $\sim$ 24.6%.**

I. Introduction

With a fast urbanization process in the past few decades, more than half of the world population are living in cities now, and this ratio is expected to achieve 66% in 2050 [1]. As a result, to ensure the efficiency, sustainability and safety of urban communities, a future city is required to manage its public services such as healthcare, transportation, education, and utilities in a more intelligent way. At the same time, the recent technology developments in Internet of Things (IoT) and Artificial Intelligence (AI) motivate the concept of smart city, that aims at making use of advanced information and communication technologies to improve the efficiency of city management. In the vision of smart city, many kinds of social infrastructures are equipped with smart sensors to enhance their productivities and functionalities, e.g., garbage cans can monitor their trash level and send request for cleaning when they are full, and storm drains can alert their maintainers in advance when a signal of overflow or obstruction emerges. Other smart city applications are found in the fields of intelligent transportation system, electricity facility, contamination monitoring, and sanitary engineering [2, 3]. It is estimated that smart city related technologies will become a massive economic engine in the coming decade, and will worth a cumulative 1.565 trillion dollars by 2020, 3.3 trillion dollars by 2025 [4].

From the previously described application scenarios, it can be observed that efficient data gathering and dissemination plays a key role in smart city construction, i.e., how to deliver the data generated by the sensors geo-distributed in city to their concerned users like city agencies, commercial service providers and citizens. Real-time communication via infrastructure-based cellular networks could be a straightforward solution. It seems that interconnecting a sensor to the public internet or to any other IP-based network infrastructure may be trivial from a technical point of view. However, it is undoubtedly difficult from an economical point of view to either equip every sensing device with a SIM card or deploy so many sensor gateways and femtocells to ensure the transmission of "Giant IoT Data" that would far beyond the so-called "Big Data" in the cyber world today [5].

In reality, many smart city applications only need data every once in a while and are delay-tolerant to the data delivery process, e.g., the maintenance of garbage cans, storm drains, and street lights [6], and the geo-advertising via vehicle ad-hoc networks (VANETs) [7]. Thus, many researchers are suggesting using data mules to reduce the cost of data delivery [6–10]. More specifically, data mules are mobile IoT nodes moving around a city and can opportunistically communicate with geo-distributed sensors via short-range wireless communication to pick up sensing data. When data mules pass by a road side unit (RSU) for data gathering, their buffered data are offloaded to the RSU via short-range wireless communication and the RSU will either forward these data to remote data centers or advertise these data to local end users. Many objects can be used as the data mules in city, such as vehicles, drones, and pedestrians holding smart phones. Among them, vehicles have attracted the most attention from both research and industrial communities, because of their steady power-supply and powerful on-board processing units. Even if it might not be realistic to involve all urban vehicles in a short term, Bonola et al. [6] have proved that, with a small-scale taxi fleet of 120 taxis in Rome, Italy, sensing data can be delivered to the RSUs deployed in the center area of city with a probability of 0.8 in one hour, and can be delivered to 80% RSUs deployed in the city within less than 24 hours. Bonola et al. call these taxis as oblivious data mules since they accomplish data delivery without any enforcement of varying their driving habits and daily business works.

As illustrated in Figure 1, the knowledge of future vehicular trajectory is essential to implement efficient vehicle-based data delivery in smart city. However, there is a great deal of uncertainty associated with vehicle mobility, since they move at their own wills. It is difficult to gain a complete knowledge about future vehicle mobility, i.e., the position of a vehicle at a given future time point. To avoid this complexity, most existing works either adopt some easy vehicle mobility patterns such as the spatial and inter-meeting time distributions to support coarse-grained vehicle mobility predictions [8, 11–13], or simplify the problem to a low-order Markov model that highly limits the quality of prediction [10]. Thus, this paper proposes a deep recurrent neural network (RNN)-based vehicle mobility prediction algorithm to facilitate sensing data delivery in smart city. To the best of our knowledge, this is the first trial of applying deep learning technology to predict vehicle mobility worldwide.

The rest of this paper is organized as follows: Section

978-1-5386-9428-2/18/$31.00 ©2018 IEEE

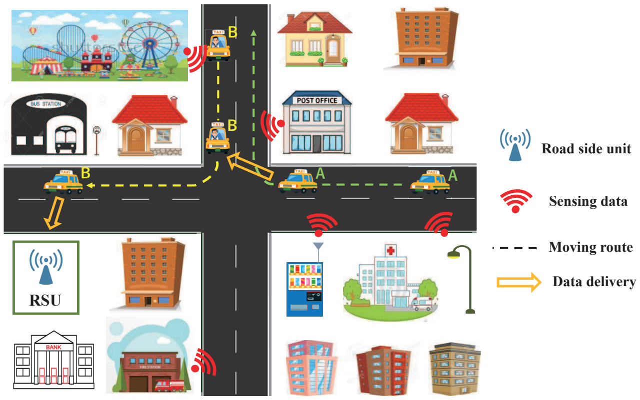


Fig. 1: The essential role of vehicle mobility prediction in the process of sensing data delivery in smart city. Assume that vehicles *A* and *B* are carrying sensing data and want to deliver data to a specific road side unit (RSU) in the left of figure. When they are contacting with each other, vehicle *A* can relay its data to vehicle *B* if vehicle *B* is predicted to contact the RSU sooner than vehicle *A*. This strategy reduces the delay of data delivery without much cost.

II reviews the previous related works. Section III defines system model and formulates the problem of vehicle mobility prediction. Section IV describes the proposed algorithm in detail. Section V presents the evaluation results of our proposal and its comparisons with the state-of-art algorithms. Finally, conclusions are drawn and future work is discussed in the last section.

II. Related Work

There are already several works have been proposed to discuss vehicle-based sensing data delivery in smart city [6, 8–10]. Bonola et al. assessed the feasibility of an oblivious data mule application in which 120 taxis were used as mobile nodes to gather and disseminate sensing data in Rome, Italy [6]. Their results have suggested that even a relatively small number of data mules can cover 80% areas of a very large and irregular city like Rome in less than 24 hours. However, their work does not introduce any concrete method to put this vision into practice. Lin et al. proposed a data delivery framework based on the vehicle-to-vehicle and vehicle-to-infrastructure communications in smart city [8]. Their algorithm derives the regular routes of vehicles from their mobility history, and estimates the contact durations between vehicles and RSUs accordingly. By ranking the contact durations between different vehicles and RSUs, they allow a vehicle having less contact duration with an RSU to opportunistically relay data to another vehicle having a longer contact duration. Their results have validated that this strategy greatly reduces the delay of sensing data delivery in smart city. Different from our proposal in this paper that aims at predicting vehicle mobility, Lin et al. assumed that vehicle mobility is almost regular every day and their algorithm fails to work when a vehicle deviates from its regular route. Although their assumption might be true for buses or a few private cars, it fails to reflect the real situations of most urban vehicles. Tang et al. described a vehicle-based data delivery scheme through the opportunistic communications in VANETs [9]. They proposed a simulated annealing algorithm to assign different kinds of sensing data with different priorities. Vehicles can discard data with low priority when their buffers overflow, and this strategy improves the success ratio of delivering data with high priority. However, this work mainly aims at coping with the storage limitations of data mules without considering their mobility patterns. The work conducted by Zhu et al. [10] is the most similar one to our proposal. Zhu et al. have validated that the future mobility of a vehicle is greatly related to its previous trajectory, and adopted a multiple order Markov model to predict vehicle mobility. Since the computational complexity of Markov model exponentially increases with the length of used vehicular trajectory, they only used a 2-order Markov model to predict vehicle mobility and that greatly limits the quality of mobility prediction. Conversely, our work in this paper adopts a deep RNN architecture that is good at extracting valuable information from a long vehicular trajectory, and the evaluation results presented in Sect. V have validated the superiority of our proposal over the Markov model-based algorithms.

III. System Model and Problem Formulation

This paper aims at designing an algorithm that can accurately predict the probability of a vehicle to enter any city area in a prediction period, given its previous mobility trajectory. As illustrated in Figure 1, when a vehicle wants to deliver its stored sensing data to an RSU, it can estimate its probability to enter the city area where that RSU is deployed. Thus, when two vehicles contact with each other opportunistically, the vehicle with a lower probability to contact that RSU in future may relay its sensing data to the one with a higher probability. This kind of data exchange based on vehicle mobility prediction can speed up sensing data delivery economically to support a number of smart city applications.

In this paper, the city space (S) of vehicle mobility prediction is discretized into a set of square grids as

$$S = \{g_1, g_2, g_3 \cdots, g_n \mid g_i \cap g_j = \emptyset\}, \tag{1}$$

where g_i indicates the identity of grid and n is the total number of grids in the city. The typical size of grid is a few hundred meters, so that when a vehicle enters a grid it can communicate with the RSUs located in the same grid via short-range wireless communication technologies like IEEE 802.11p [14] and Wi-SUN [15] that cover a transmission range of several hundred meters. The time is also slotted so that the instant position of a vehicle is sampled in every time slot by using its positioning system like GPS.

When a vehicle v is moving in the city, its position at a given time slot t is denoted by a random variable s_t that takes a value from the city space defined by Eq. (1). Hence, its previous trajectory at t is given by

$$T_t^k = \langle s_{t-k+1}, s_{t-k+2} \cdots s_{t-1}, s_t \rangle, \tag{2}$$

where s_i indicates the grid where the vehicle v locates at time slot i, and k represents the order of trajectory. Given the k-order trajectory of the vehicle v at time slot t, its probability to be in a grid g_i at a future time slot $t^{'}$ can be represented by

$$P_{t'}(g_i) = \{P(s_{t'} = g_i \mid T_t^k), t^{'} > t\}. \tag{3}$$

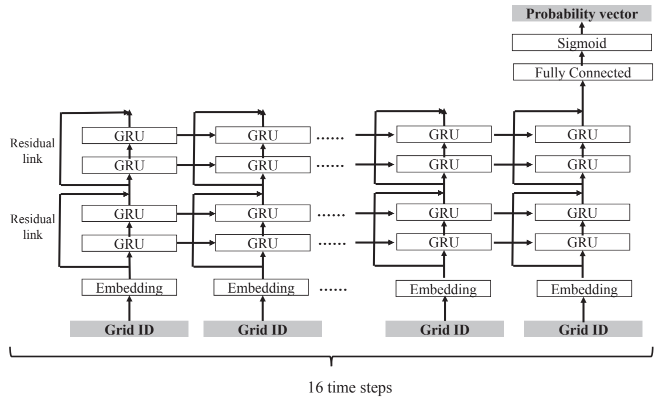


Fig. 2: The neural network architecture of the proposed algorithm.

Finally, given its k-order trajectory at time slot t, the probability for v to pass through g_i in a future prediction period $(t, t+c]$ can be denoted by

$$P_{t\sim t+c}(g_i) = P(\cup_{j=1}^{c}\ s_{t+j} = g_i \mid T_t^k), \tag{4}$$

where c is the length of prediction period. The objective of this paper is to design a vehicle mobility prediction algorithm that can solve Eq. (4) accurately and efficiently.

IV. The Proposed Algorithm to Predict Vehicle Mobility

This section describes our proposed deep RNN-based algorithm for vehicle mobility prediction. Benefitted from the capability of RNN to process long sequential data, a 16-order vehicular trajectory is used to predict vehicle mobility and that is much longer than the one utilized by conventional algorithms. The choice of 16-order is kind of arbitrary, and it is a trade-off between our computational resources and algorithm performance. Figure 2 depicts the neural network architecture of our proposal.

In this algorithm, every grid identity is encoded by an n-dimensional one-hot vector where n is the total amount of grids in the city space, e.g., a grid identity of 3 is represented by $(0, 0, 0, 1, 0 \cdots 0)$. Every data item includes 16 grid identities that indicate where the corresponding vehicle positioned in the previous 16 time slots. These sparse grid identities are first mapped into a smaller-and-denser vector by an embedding layer of RNN. Briefly speaking, embedding method is a feature learning technique that is widely used in deep learning [16, 17]. By compacting a large-but-sparse vector to a small-and-dense one, this method is helpful to accelerate the training process of neural network without much performance penalty.

The output vector of embedding layer is fed into the Gated Recurrent Unit (GRU) cells [18] of this RNN. There are two GRU blocks in the RNN, and every GRU block comprises two layers of GRU cells. When processing the embedded vector of vehicular trajectory, these GRU cells can memorize the long-term spatial-temporal correlations existing in the training data items and utilize these correlations to predict vehicle mobility. To deal with the issues of gradient vanishing in training a deep neural network, a residual link [19] is supplemented to bridge the input and output of every GRU block. A fully connected layer and a sigmoid layer map the output vector of the last GRU cell to a probability vector denoted by

$$P_{predict}^{j} = (p_{g_1}^j, p_{g_2}^j, p_{g_3}^j, \cdots p_{g_{n-1}}^j, p_{g_n}^j), \tag{5}$$

where j indicates the index number of data item, and $p_{g_i}^j$ represents the conditional probability for the corresponding vehicle of this data item to pass through the grid g_i given its 16-order trajectory.

A binary cross-entropy loss function is adopted to train this RNN. More specifically, the ground truth vector of a data item indexed by j can be represented by a binary vector

$$L_{truth}^{j} = (l_{g_1}^j, l_{g_2}^j, l_{g_3}^j \cdots, l_{g_{n-1}}^j, l_{g_n}^j), \tag{6}$$

where $l_{g_i}^j = 1$ when the vehicle actually passes through the grid g_i during prediction period in reality, and $l_{g_i}^j = 0$ otherwise. By convention, $l_{g_i}^j$ is called a positive label of L_{truth}^j when it is one and a negative label otherwise. The binary cross-entropy loss of a dataset is given by

$$C = -\frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} l_{g_i}^j * log(p_{g_i}^j), \tag{7}$$

where m is the total number of data items in the dataset, and n is the total amount of grids in the city space.

Finally, there are two characteristics of the proposed algorithm need to be mentioned: (1) Since a vehicle may pass through multiple grids during a prediction period, there would be multiple positive labels (i.e., multiple ones) in the ground truth vector L_{truth} of every data item. This kind of prediction task is called multi-label or multi-task prediction in many literatures; (2) In a usual vehicle mobility prediction period like 5 or 10 minutes, the number of different grids that a vehicle can pass through is quite limited, compared with the whole city space. Thus, there are usually many negative labels but a few positive labels in the ground truth vector L_{truth} of every data item, i.e., the concerned label distribution of prediction model is highly-skewed. Clearly, both characteristics increase the difficulty of predicting vehicle mobility, since a prediction algorithm has to pick a few but multiple positive labels from a lot of negative labels accurately. To the best of our knowledge, our proposal is the first trial to evaluate the capability of deep RNN on this kind of difficult tasks in the background of spatio-temporal correlated GPS data.

V. Performance Evaluations

This section presents the evaluation results of our proposed algorithm. A taxi mobility dataset that is obtained from our smart city testbed deployed in Tokyo, Japan [20] is used in the following evaluations. This dataset includes the mobility trajectories of 65 taxis within a duration of 4 months, i.e., from January 2018 to April 2018. In a time slot of 1 minute, every taxi periodically send its instant position to a data center. The whole urban area of Tokyo which covers about 700 square kilometers is treated as the city space for mobility prediction, and it is divided into 2791 square grids of 500×500 meters. This whole dataset is further divided into training, hold-out validation, and test datasets. The training dataset contains the vehicular trajectories in first 14 weeks ($01/01/2018 \sim 08/04/2018$), the hold-out validation dataset contains the vehicular trajectories in the next 1 week

TABLE I: Basic evaluation parameters.

Development framework	Tensorflow r1.8
Embedding layer size	200
GRU cell size	200
GRU cell activation function	exponential linear unit (ELU)
Fully connected layer size	2791
Sigmoid layer size	2791
Drop out ratio	0.5
Learning rate	1e-3
Learning rate decay	0.99 per epoch
Prediction period	10 minutes
Time slot	1 minute
Grid size	500 m × 500 m
Grid amount	2791

(09/04/2018 $\sim$ 15/04/2018), and the test dataset contains the vehicular trajectories in final 2 weeks (16/04/2018 $\sim$ 30/04/2018). The proposed algorithm is implemented by using Tensorflow r1.8 [21] and the basic parameters used in the following evaluation are summarized in Table I.

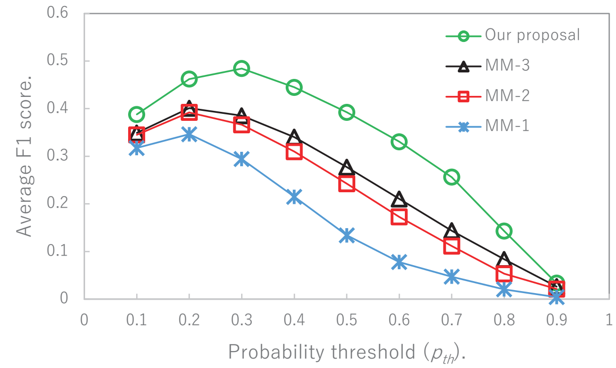
The remaning part of this section presents the comparison results of the proposed algorithm and the state-of-art Markov model-based prediction algorithms proposed by Zhu et al. [10]. Instead of only using the 2-order Markov model-based algorithm (MM-2) recommended in [10], both 1-order and 3-order Markov model-based algorithms (MM-1 and MM-3) are also evaluated to better explore evaluation results. In the following evaluations, these algorithms are compared based on three metrics, i.e., precision, recall, and F1 score. These metrics require a binary prediction vector for every data item j, while the output of our proposed algorithm and Markov model-based algorithms is a probability prediction vector P^j_{pred} as shown by Eq. (5). Consequently, a threshold $p_{th} \in [0, 1]$ is used for transforming P^j_{pred} into a binary prediction vector,

$$P^j_{b-pred} = (b^j_{g_1}, b^j_{g_2}, b^j_{g_3}, \cdots b^j_{g_{n-1}}, b^j_{g_n}), \tag{8}$$

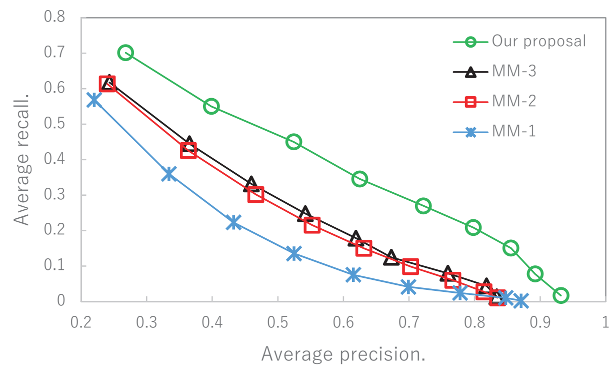
where $b^j_{g_i} = 1$ if $p^j_{g_i} \in P^j_{pred}$ is no less than p_{th} and $b^j_{g_i} = 0$ otherwise. Intuitively, $b^j_{g_i} = 1$ indicates that the corresponding vehicle of data item j is predicted to pass through the grid g_i since its corresponding probability is no less than the given threshold, and vice versa. Based on the definitions of binary prediction vector P^j_{b-pred} in Eq. (8) and ground truth vector L^j_{truth} in Eq. (6), a brief description of the previous mentioned four metrics are given below and their more detailed definitions are available in [22]:

Precision (p): The precision is defined for a data item j as the size of the intersection set of its positive labels in P^j_{b-pred} and its positive labels in L^j_{truth} divided by the size of the set of its positive labels in P^j_{b-pred}. A higher precision value denotes that there is less false positive predictions;

Recall (r): The recall is defined for a data item j as the size of the intersection set of its positive labels in P^j_{b-pred} and its positive labels in L^j_{truth} divided by the size of the set of its positive labels in L^j_{truth}. A higher recall value denotes that there is less false negative predictions;



(a) The average F1 scores of algorithms.



(b) The average precisions and recalls of algorithms.

Fig. 3: The average F1 scores, precisions, and recalls of different algorithms with a prediction period of 10 minutes.

F1 score ($F1$)**:** Since there is a trade-off between the values of precision and recall, F1 score is defined as their harmonic mean to give an integrated metric for comparing different algorithms, i.e., $F1 = (2 \times p \times r)/(p + r)$. A higher F1 score denotes a better quality of vehicle mobility prediction;

Figure 3(a) shows the average prediction F1 scores of different algorithms with a prediction period of 10 minutes. The performances of Markov model-based algorithms increase with the orders of their adopting vehicular trajectory. For example, compared with MM-1 that only adopts a 1-order vehicular trajectory in its prediction, MM-2 and MM-3 increase their highest F1 scores by 13.2% and 15.9%. Since the proposed algorithm can utilize a much longer (i.e., 16-order) vehicular trajectory in its prediction than all Markov model-based algorithms, it improves the highest F1 score of the best performed MM-3 by 20.8%. This result proves that a long trajectory of vehicle mobility can significantly improve the quality of vehicle mobility prediction.

The F1 scores of different algorithms also vary with the threshold p_{th} that is used by Eq. (8) to transform a probability prediction vector to a binary prediction vector. The reason of this phenomenon is further illustrated in Figure 3(b). With a low probability threshold, these algorithms tend to predict that a vehicle will pass through a grid (i.e., with a prediction

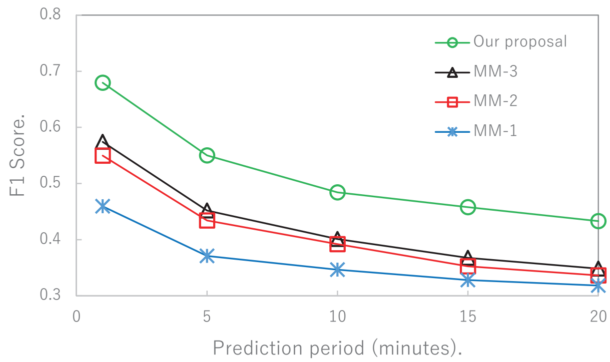


Fig. 4: The highest F1-Scores of different algorithms with a variant prediction period of 1, 5, 10, 15, and 20 minutes.

value of one) even if its corresponding probability is low. This strategy reduces false negative predictions to produce a high recall value, while increasing false positive predictions to cause a low precision value. Conversely, a high threshold would lead to a low recall value but a high precision value. Since F1 score is the harmonic mean of precision and recall values, it achieves its highest value only when a threshold well balances precision and recall. As a result, to avoid the ambiguity of comparing the F1 scores of different algorithms with different probability thresholds, an algorithm's highest F1 score is a well-accepted metric to represent its prediction capability.

Figure 4 shows the highest prediction F1 scores of different algorithms with a variant prediction period of 1, 5, 10, 15, and 20 minutes. Again, the performances of Markov model-based algorithms increase with the orders of their adopting vehicular trajectory. For example, compared with MM-1, the best-performed Markov model-based algorithm MM-3 increases its highest F1 scores by a range of 9.3% $\sim$ 25.0% with different prediction periods. Benefitted from its capability of using a long vehicular trajectory, our proposal further improves the highest F1 score of MM-3 by a range of 18.3% $\sim$ 24.6%. In addition, the improvements of our algorithm become more significant with the increase of prediction period, e.g., it improves the highest F1 score of MM-3 by 18.3% when the prediction period is 1 minute, while improving that by 24.6% and 24.3% when the prediction periods are 15 and 20 minutes. This trend shows that a long vehicular trajectory can improve the quality of vehicle mobility prediction, especially when the concerned prediction period becomes long.

VI. Conclusions and Discussions

This paper proposes a novel vehicle mobility prediction algorithm to facilitate vehicle-based sensing data delivery in smart city. The proposed algorithm adopts a deep recurrent neural network architecture and this strategy makes it capable of retrieving valuable information from a much longer vehicular trajectory than any other existing algorithm. Extensive evaluation results have validated that our proposal significantly improves the quality of vehicle mobility prediction compared with the state-of-art algorithms.

Since this is the first proposal of applying deep learning technology to predict vehicle mobility, it can still be further discussed from several aspects:

(1) Centralized or distributed learning strategy: our algorithm adopts a centralized learning strategy, i.e., it gathers vehicles' position data in a data center and process them by a deep learning algorithm. Our evaluations in this paper have validated that an RNN model trained in several hours can be used to predict vehicle mobility in a future period of 2 weeks, and its model size is only about 20 megabytes. Thus, it is realistic for a smart vehicle to retain a pre-trained prediction model locally and update the model from data center periodically. Conversely, a distributed learning strategy aims at training a prediction model by using the on-board processing unit and mobility trajectory of each vehicle separately. There are pros and cons for both strategies, e.g., the centralized strategy can explore the collaborative mobility knowledge from a large number of vehicles by using powerful computational resources, while the distributed strategy may be good at discovering the unique mobility pattern of different vehicles even with limited computational resources. To the best of our knowledge, there is no work clarifies the trade-off between these two strategies quantitatively. It is also interesting to consider the feasibility of a hybrid framework that can manage these two strategies simultaneously and coherently.

(2) Different application scenarios and parameters: This paper proposes a vehicle mobility prediction algorithm in the background of vehicle-based sensing data delivery. However, vehicle mobility prediction technology may also help other smart city applications like traffic flow prediction and vehicle scheduling for online car-hailing services. Different application scenarios may require different parameter settings such as the range of concerned city space, the grid size of city space and the sampling rate of vehicle position. Some of these parameters are further correlated and have to be adjusted dependently, e.g., a smaller grid size requires a higher sampling rate of vehicle position to avoid missing too much information of vehicular trajectory during a sampling interval. It is necessary to tune the hyper-parameters of our algorithm such as its order of vehicular trajectory and size of GRU cells to fit new application parameters. However, the key empirical findings of this research work never change with these application parameters, i.e., a long vehicular trajectory is crucial to predict vehicle mobility, and the proposed algorithm can improve the quality of vehicle mobility prediction by processing a long vehicular trajectory efficiently. Thus, it is an interesting future work to customize this algorithm for different smart city application scenarios.

References

[1] UN DESA, "World's population increasingly urban with more than half living in urban areas," *UN News Center*, 2014.

[2] Martin Strohbach, Holger Ziekow, Vangelis Gazis, and Navot Akiva, "Towards a big data analytics framework for iot and smart city applications," in *Modeling and processing for next-generation big-data technologies*, pp. 257–282. Springer, 2015.

[3] Nguyen Cong Luong, Dinh Thai Hoang, Ping Wang, Dusit Niyato, Dong In Kim, and Zhu Han, "Data collection and wireless communication in internet of things (iot) using economic analysis and pricing models: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2546–2590, 2016.

[4] Bo Tang, Zhen Chen, Gerald Hefferman, Shuyi Pei, Tao Wei, Haibo He, and Qing Yang, "Incorporating intelligence in fog computing for big data analysis in smart cities," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2140–2150, 2017.

[5] Mung Chiang and Tao Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[6] Marco Bonola, Lorenzo Bracciale, Pierpaolo Loreti, Raul Amici, Antonello Rabuffi, and Giuseppe Bianchi, "Opportunistic communication in smart city: Experimental insight with small-scale taxi fleets as data carriers," *Ad Hoc Networks*, vol. 43, pp. 43–55, 2016.

[7] Jianping He, Lin Cai, Peng Cheng, and Jianping Pan, "Delay minimization for data dissemination in large-scale vanets with buses and taxis," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1939–1950, 2016.

[8] Zhijie Lin, Yongxuan Lai, Xing Gao, Guilin Li, Tian Wang, and Guohua Huang, "Data gathering in urban vehicular network based on daily movement patterns," in *Computer Science & Education (ICCSE), 2016 11th International Conference on*. IEEE, 2016, pp. 641–646.

[9] Zhipeng Tang, Anfeng Liu, and Changqin Huang, "Social-aware data collection scheme through opportunistic communication in vehicular mobile networks," *IEEE Access*, vol. 4, pp. 6480–6502, 2016.

[10] Yanmin Zhu, Yuchen Wu, and Bo Li, "Trajectory improves data delivery in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1089–1100, 2014.

[11] Wei Liu, Ryoichi Shinkuma, and Tatsuro Takahashi, "Opportunistic resource sharing in mobile cloud computing," *IEICE Transactions on Communications*, vol. 97, no. 12, pp. 2668–2679, 2014.

[12] Kyunghan Lee, Yung Yi, Jaeseong Jeong, Hyungsuk Won, Injong Rhee, and Song Chong, "Max-contribution: On optimal resource allocation in delay tolerant networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[13] Jérémie Leguay, Timur Friedman, and Vania Conan, "Dtn routing in a mobility pattern space," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. ACM, 2005, pp. 276–283.

[14] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap, "Multihop-cluster-based ieee 802.11 p and lte hybrid architecture for vanet safety message dissemination," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2621–2636, 2016.

[15] Hiroshi Harada, Keiichi Mizutani, Jun Fujiwara, Kentaro Mochizuki, Kentaro Obata, and Ryota Okumura, "Ieee 802.15. 4g based wi-sun communication systems," *IEICE Transactions on Communications*, vol. 100, no. 7, pp. 1032–1043, 2017.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[17] Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding.," in *Interspeech*, 2013, pp. 3771–3775.

[18] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] Wei Liu, Kiyohide Nakauchi, and Yozo Shoji, "A neighbor-based probabilistic broadcast protocol for data dissemination in mobile iot networks," *IEEE Access*, 2018.

[21] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[22] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern recognition*, vol. 45, no. 9, pp. 3084–3104, 2012.