

The 10th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 29 - May 2, 2019, Leuven, Belgium

Attention-based Recurrent Neural Network for Urban Vehicle Trajectory Prediction

Seongjin Choi^a, Jiwon Kim^b, Hwasoo Yeo^{a,c,*}

^aKorea Advanced Institute of Science and Technology, 291 Daehak-ro, Daejeon 34141, Republic of Korea

^bThe University of Queensland, St. Lucia 4072, Australia

^cNew York University Abu Dhabi, 129188, Saadiyat Island, Abu Dhabi, United Arab Emirates

Abstract

As the number of various positioning sensors and location-based devices increase, a huge amount of spatial and temporal information data is collected and accumulated. These data are expressed as trajectory data by connecting the data points in chronological sequence, and these data contain movement information of any moving object. Particularly, in this study, urban vehicle trajectory prediction is studied using trajectory data of vehicles in urban traffic network. In the previous work, Recurrent Neural Network model for urban vehicle trajectory prediction is proposed. For the further improvement of the model, in this study, we propose Attention-based Recurrent Neural Network model for urban vehicle trajectory prediction. In this proposed model, we use attention mechanism to incorporate network traffic state data into urban vehicle trajectory prediction. The model is evaluated by using the Bluetooth data collected in Brisbane, Australia, which contains the movement information of private vehicles. The performance of the model is evaluated with 5 metrics, which are BLEU-1, BLEU-2, BLEU-3, BLEU-4, and METEOR. The result shows that ARNN model have better performance compared to RNN model.

© 2019 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Vehicle Trajectory; Trajectory Prediction; Recurrent Neural Network; Attention Mechanism; Network Traffic State

1. Introduction

Large amount of location data collected with various location sensors and location-aware devices are studied in the form of trajectory data. Trajectory data is a trace of an object that moves in geographical spaces. This is represented by a chronologically ordered sequence of location [1]. In this study, we focus on one type of trajectory data, which is urban vehicle trajectory data. The urban vehicle trajectory data is one type of trajectory data that represents the movements of vehicle in urban networks. This data gives opportunities to understanding movement patterns of urban

* Corresponding author. Tel.: +82-42-350-5674

E-mail address: hwasoo@kaist.ac.kr

traffic networks by offer a large information about aggregate flows and disaggregate travel behaviors including user-centric traffic experience, and mobility patterns of whole system. Previous studies have used these data to perform travel pattern analysis [2, 3] and develop real-world applications such as trajectory-based route recommendation system [4] and trajectory-based bus arrival prediction [5]. Of the particular interest, this study analyzes the large amounts of trajectories of people and vehicles in city by solving trajectory-based location prediction problem, which makes predictions on next locations [6, 7], destinations [8, 9, 10, 11, 12, 13], or the occurrences of traffic related events [14].

Among many applications of trajectory data mining [15], this study focuses on trajectory-based location prediction problem. This problem concerns analyzing large amounts of trajectories of people and vehicles moving around a city to make predictions on their next locations [6, 7, 8], destinations [9, 10, 11, 12, 13], or the occurrences of traffic related events such as traffic jams and incidents [14]. In this study, we address the problem of predicting the sequence of next locations that the subject vehicle would visit, based on the information on the previous locations from the origin of the current trip and historical database representing the urban mobility patterns.

In the previous study [16], a methodology which predicts next locations of vehicle by using Recurrent Neural Network (RNN) model is proposed. RNN is a neural network model, which is widely used in natural language processing. In the previous study, we shed light on the similarity of text generation and trajectory based location prediction, and adopted RNN model for trajectory based location prediction. The previously visited location data was the only input to RNN model [16]. In spite of simple structure of model, RNN based location prediction model produced successful results. For example, the probability of predicting success about next location of vehicle was greater than 0.7 for more than 50% of all the tested trajectory samples, while the base case model showed accuracy of less than 5% of the samples [16].

For the improvement of location prediction accuracy, this study suggests novel methodology on existing RNN model, which has additional input that help predictions by allowing the incorporation of heterogeneous input sources. We considered the traffic conditions of urban traffic network at the start of the journey as a specific input in the study. These days, drivers can easily observe the current traffic state in the urban traffic networks and select their route by using various traffic information and routing services [17]. Therefore, these routes of each vehicles are expected to be effected by traffic conditions in networks at the start of their journey. Based on this idea, this study suggests an Attention-based RNN model for location sequence prediction that enable the RNN model to consider current traffic conditions as an additional input. A detailed explanation is in the Methodology section.

2. Methodology

Vehicle trajectory which consists of l number of data points of longitude (x) and latitude (y) is represented as $Tr = [(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)]$. Vehicle trajectory, Tr , can be represented as a sequence of cells $[c_1, c_2, \dots, c_m]$ if the urban traffic network is partitioned into several cells $(c_i, i \in 1, \dots, m)$. Since each cells can cover multiple data points, length of cell sequence m is always less than or equal to length of original trajectory sequence l (i.e. $m \leq l$). Two

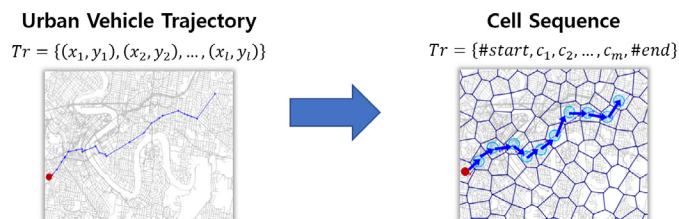


Fig. 1. Representing urban vehicle trajectory as cell sequence

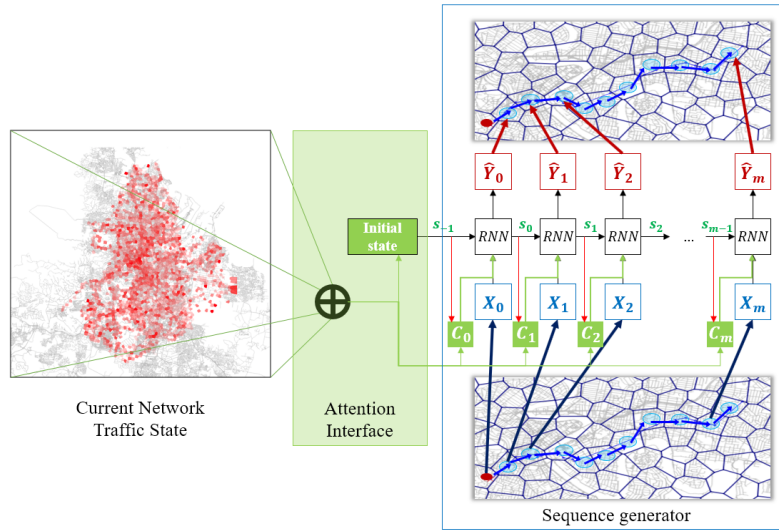


Fig. 2. Structure of the proposed Attention-based Recurrent Neural Network model (ARNN) for cell sequence prediction

virtual tokens *#start* and *#end* are added each front and back of cell sequence to represent the start and end of the trip. For RNN training, validating and testing, this cell sequence is separated to two parts, *X* and *Y*, which are input vector

$$\begin{aligned} X &= [X_0, X_1, X_2, \dots, X_m] \equiv [\text{\#start}, c_1, c_2, \dots, c_m] \\ Y &= [Y_0, Y_1, Y_2, \dots, Y_m] \equiv [c_1, c_2, \dots, c_m, \text{\#end}] \end{aligned} \quad (1)$$

RNN model for predicting cell sequence was developed and evaluated in previous study [16], but adding additional inputs to RNN model is not a simple task. Since RNN model is suited for processing sequential data that considers dependency across time or sequence steps, when additional input is sequential data, adding another sequence input can be done with simple extension. RNN model can have multiple input layers and various hidden features to combine multiple sequence inputs to compute the output. Otherwise, it needs to be processed outside the RNN model. We are going to use non-sequential traffic state information as an additional input. As a result, an additional structure to incorporate this non-sequential traffic state information is necessary.

The *attention mechanism* is one way to address this challenge. The attention mechanism is an interface between sequential inputs processed inside the RNN model and external processed information, as illustrated in Fig 1. This mechanism in neural networks was first introduced to imitate the attention mechanism in the human brain and has shown great improvement in performance of applications such as machine language translation [18] and video captioning [19]. The cell sequence prediction model or cell sequence generator that used the attention mechanism can focus on specific parts of the network traffic state input and use information for cell sequence generation. Two tasks given to the attention mechanism are as follows: setting initial state for the RNN and providing network-wide traffic state information at each cell generation step. Usually, the initial state vector of RNN cell is set as zero vector because there are not additional inputs in basic form of RNN model. However, ARNN has an additional information of network traffic state data that should be embedded into the model. In addition, the attention mechanism allows the RNN to consider traffic state in each step of prediction by calculating attention weights and the context vector.

The drivers can easily obtain the current traffic state in the urban traffic networks, and plan their journey by using various traffic information and routing services [17, 20]. For example, between routes A and B, a driver is likely to choose route A when route B is congested and vice versa. As a result, the location sequences (chosen routes) of individual vehicles are expected to be influenced by the network traffic conditions at the beginning of their journeys. It is thus desirable to incorporate network-wide traffic state information and route choice behavior depending on the prevailing traffic state into the RNN-based cell sequence prediction model to increase the models prediction accuracy.

Fig. 2 shows the structure of the Attention-based Recurrent Neural Network (ARNN) model for the cell sequence prediction. There are two types of input data in this model: the first is the network traffic state data and the second

is the cell sequence representation of vehicle trajectory data. The model first processes the current network traffic state and calculates the initial state (s_{-1}) for the RNN unit. Then, the attention interface calculates the context vector (C_i) based on the previous state vector. The context vector (C_i) is used as input to the i^{th} RNN unit as well as the corresponding input vector element (X_i) to update current state vector (s_i). The attention weight $\alpha(i, j)$ is calculated based on the context vector and previous state vector ($\alpha(i, j) = f(C_i, s_{i-1})$). The attention weight represents the probability to attend to j^{th} cell at i^{th} sequence. Therefore, the sum of $\alpha(i, j)$ at each sequence is 1 ($\sum_j \alpha(i, j) = 1$).

The input cell sequence (X) is processed based on the word-embedding method to represent the hidden features of the cells. In the training step, input vector X is directly used as an input of each RNN unit in order to calculate the output vector (\hat{Y}_i). However, in the testing step, only the front n cell sequence elements are directly used. Afterward, since the output vector represents the probability of each cell being visited, we use a random sampling based on the multinomial distribution with probability \hat{Y}_i to extract the next cell, also it is used as the next input vector element.

A basic Long Short Term Memory (LSTM) cell [21] is used as RNN cell. And the model also uses the Adam optimizer to update the model parameters [22].

3. Model Performance Evaluation

3.1. Data

3.1.1. Urban Vehicle Trajectory Data

The vehicle trajectory data used in this research are collected from the Bluetooth sensors in Brisbane, Australia, provided by Queensland Department of Transport and Main Roads (TMR) and Brisbane City Council (BCC). The Bluetooth sensors are installed in state-controlled roads and intersections located inside the Brisbane City, and they detect Bluetooth devices (e.g., in-vehicle navigation systems and mobile devices) passing the sensors and record their passage time. By connecting the data points containing the same identifier of the Bluetooth device (MAC ID), the vehicle trajectories of individual vehicles can be constructed. Each vehicle trajectory represents a time-ordered sequence of Bluetooth sensor locations that a subject vehicle passes. If the corresponding vehicle does not move for more than an hour, it is considered that the vehicle trip has terminated. For this case study, we used the vehicle trajectory data collected in March 2016. There are approximately 276,000 trajectories in one day, and a total of 8,556,767 vehicle trajectories are collected in March 2016. We randomly sampled 200,000 vehicle trajectories for the training dataset, 10,000 vehicle trajectories for the validation dataset (used in hyper-parameter searching), and 200,000 vehicle trajectories for the testing dataset.

Brisbane urban traffic network is divided into cells to use the vehicle trajectory clustering and cell partitioning method proposed in the previous research [23, 24]. The desired radius of the cells is set to be 300m. Accordingly, a total of 5,712 cells are generated. Among them, 2,746 cells are considered to be active since the rest of the cells are not visited by any vehicles in the historical data of vehicle trajectories. The vehicle trajectory data are processed and transformed into cell sequence data.

3.1.2. Network Traffic State Data

There are several ways to represent the network traffic state such as density and average speed. In this study, vehicle accumulation, which is understood as the density of cells, is used to represent the network traffic state. The vehicle accumulation for a given cell is estimated by counting the number of vehicles that are present within the cell at a given instant point in time. We processed the vehicle trajectory data and calculated the vehicle accumulation of each cell at each minute. The vehicle accumulation data are normalized by dividing the vehicle accumulation by the historical maximum number of vehicle accumulation in each cell.

The vehicle accumulation data are used as the network traffic state input to the ARNN model. When the ARNN model is trained through each cell sequence, the model receives the vehicle accumulation data on a whole network from 10 minutes before the start time of the sequence. As a result, the shape of the input vehicle accumulation data is $[N, 10]$, where N is the number of cells in the study network.

3.2. Score based Evaluation of Generated Cell Sequences

In this study, two widely used evaluation metrics in sequence modeling, BLEU score and METEOR score, are used to evaluate the performance of the models (RNN and ARNN). In the previous study [16], we used the complementary cumulative distribution function of the probability to measure how accurately the model predicts the next 1, 2, or 3 consecutive cells. While this measure is intuitive and easy to interpret, there is a drawback in this method in that it considers element-wide prediction accuracy and does not take into account the whole sequence. The element-wide performance measure can be sensitive to small local mis-predictions and tend to underestimate the performance of the model. For example, when the original cell sequence is $[\#start, c_1, c_2, c_3, \#end]$ and a model is asked to predict the next cells based on the given cell sequence $[\#start, c_1]$, the prediction of $[c_2, c_4, c_3, \#end]$ will be considered as incorrect and performing poorly by our previous method because the model miss-predicted one cell c_4 , even though the overall sequence is very similar to the original sequence. As such, this study employs BLEU score and METEOR score that consider the whole sequence and thus more robust and accurate as performance measure for sequence modeling.

3.2.1. BLEU score

BLEU score [25] is one of the most widely used evaluation score in sequence modeling, especially in natural language processing. *BLEU* uses modified form of precision to compare a reference sequence and a candidate sequence. For each cell in the candidate (generated) cell sequence, the algorithm takes the maximum count (m_{max}) of the cell in the reference cell sequence.

$$P = \frac{\min(m, m_{max})}{w_t} \quad (2)$$

where m refers to the number of a single cell in candidate cell sequence which are also found in the reference cell sequence, m_{max} is sum of maximum number of each cell in reference sequence which are in candidate sequence, and w_t refers to the number of cells in candidate sequence.

Basically, the *BLEU* score uses a single cell as a unit for score calculation. If the sequences are divided by multiple cells (n cells), we calculate the *BLEU* score for each n -cell-unit. This measures whether the model can predict the next cells consecutively. The *BLEU* – n score is the average of *BLEU* scores, in which the sequence is divided at all orders from 1 to n .

3.2.2. METEOR score

METEOR [26] first creates an alignment between candidate cell sequence and reference cell sequence. The alignment is a set of mappings between the most similar cells. Every cell in the candidate sequence should be mapped to zero or one cell in the reference sequence. *METEOR* chooses an alignment with the most mappings and the fewest crosses (fewer intersection between mappings).

To calculate *METEOR* score, we first define precision P and recall R .

$$P = \frac{\min(m, m_{max})}{w_t} \quad (3)$$

$$R = \frac{\min(m, m_{max})}{w_r} \quad (4)$$

where m refers to the number of single cells in candidate cell sequence which are also found in the reference cell sequence, m_{max} refers to the sum of maximum number of each cell in reference cell sequence which are in candidate cell sequence, w_t refers to the number of cells in candidate cell sequence, and w_r refers to the number of cells in reference cell sequence.

Then, we calculate the weighted harmonic mean between precision and recall, where the ratio of the weights is 1:9.

$$F_{mean} = \frac{10}{\frac{1}{P} + \frac{9}{R}} = \frac{10PR}{R+9P} \quad (5)$$

To account for congruity with respect to a longer cell segment that appears both in reference and candidate cell sequences, we generate mappings based on the longer cell segment and use it to compute the penalty p . The more

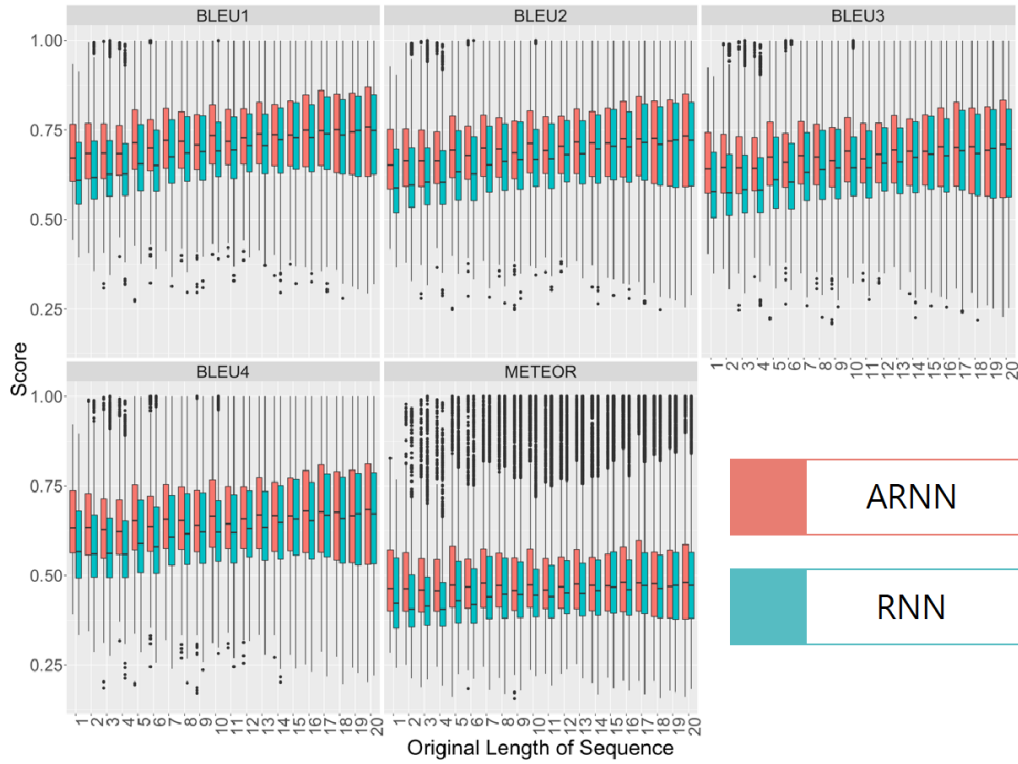


Fig. 3. Boxplot of models (ARNN, RNN) for each original length of sequence (m)

mappings there are, which are not adjacent in the reference and the candidate cell sequence, the higher the penalty will be. The penalty is calculated as follows:

$$p = 0.5\left(\frac{c}{u_m}\right)^3 \quad (6)$$

where c is a set of single cells that are not adjacent in the candidate and reference sequence, and u_m is the number of single cells that have been mapped. This penalty reduces F_{mean} up to 50% and calculate the *METEOR* score (M).

$$M = F_{mean}(1 - p) \quad (7)$$

3.3. Score Result

For each sequence in the test dataset, the scores are calculated by the following procedure.

Let Tr be the subject cell sequence with length m , which is expressed as:

$$Tr = [\#start, c_1, c_2, \dots, c_m, \#end] \quad (8)$$

The subject cell sequence is divided into 2 parts: The sequence given ($Tr_{1:g}$) and the sequence to be predicted ($Tr_{g+1:m}$), where g is the number of cells given to the models (ARNN and RNN).

$$Tr_{1:g} = [\#start, c_1, c_2, \dots, c_g] \quad Tr_{g+1:m} = [c_{(g+1)}, \dots, c_m, \#end] \quad (9)$$

Each model predicts 100 candidate cell sequences based on $Tr_{1:g}$ producing a set of 100 $Tr_{g+1:m}$ sequences for each $Tr_{1:g}$. The generated candidate cell sequences are the cell sequences that have $\#end$ token at the end, representing that the trip has terminated. These candidate cell sequences may not have the same length with the original cell sequence. The length can be longer or shorter depending on when the model predicts $\#end$ token. The $Tr_{g+1:m}$ is used as reference

cell sequence to calculate the score presented above. For each score ($BLEU - 1$, $BLEU - 2$, $BLEU - 3$, $BLEU - 4$, and $METEOR$), 100 score values are calculated based on the generated 100 candidate cell sequences. The average value of each score is used to represent the model performance of the corresponding cell sequence (Tr).

10,000 cell sequences in test dataset is used to calculate the five scores, which are $BLEU - 1$, $BLEU - 2$, $BLEU - 3$, $BLEU - 4$, and $METEOR$. Fig. 3 shows the score result of each model. The x-axis represents the original length of the sequence, and the y-axis represents the value for each score metrics. The result of ARNN model (red color) shows better performance compared to the result of RNN model (blue color).

As the original length of the sequence increases, the performance difference between the two models tends to decrease. And the ARNN model predicted a short cell sequences array to 12% more accurately. The ARNN model is superior to the RNN model in terms of both BLEU score and METEOR score. It is worth noting that the ARNN model showed improved performance on METEOR score. This is because high METEOR score require accurate description of the visiting order of cells, as well as better prediction of visited cells. Therefore, it can be seen that the ARNN model using the attention mechanism can improve both the cell configuration accuracy and the cell alignment of the sequence.

4. Conclusion and Future Studies

We proposed a new approach to incorporate network traffic state data into urban vehicle trajectory prediction model, based on the previous work. We used the attention mechanism to use the network traffic state data for vehicle trajectory prediction, and the results of the ARNN model are compared with those of the existing RNN model. As a result, the ARNN model performed better than the RNN model. We have confirmed that using attention mechanism to structurally connecting the network traffic state input to RNN model is more effective in predicting the path of the vehicle. Notably, ARNN showed significant performance improvement in the METEOR score because it considered not only the cells to be visited but also the alignment of the cells in the sequence. The performance improvement rates tend to decrease and converge to 1 as the length of original cell sequence increases. To further improve the ARNN model, this problem should be studied to keep performance improvements at a steady level.

Future research should address some limitations of this study. First, in this study, network traffic state data was normalized based on the historical maximum value of each cell. This method makes it easier to express the network traffic state data, but since the normalized cell with very low traffic is sensitive to little amount of vehicles and considered to congested, the model reacts sensitively to the cell and confuses the cell sequence prediction. Therefore, different types of normalization methods should be tested in further researches. Second, because score metrics used in this study ($BLEU$ and $METEOR$), are new in the transportation domain, and the implication of these metrics should be further investigated and proposed.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00121, Development of a Traffic Predictive Simulation SW for Improving the Urban Traffic Congestion). This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2019-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

- [1] Y. Zheng, Trajectory data mining: an overview, *ACM Transactions on Intelligent Systems and Technology (TIST)* 6 (3) (2015) 29.
- [2] J. Kim, H. S. Mahmassani, Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories, *Transportation Research Procedia* 9 (2015) 164–184.
- [3] M. Yildirimoglu, J. Kim, Identification of communities in urban mobility networks using multi-layer graphs of network traffic, *Transportation Research Procedia* 27 (2017) 1034–1041.
- [4] J. Yuan, Y. Zheng, X. Xie, G. Sun, T-Drive: Enhancing Driving Directions with Taxi Drivers' Intelligence., *IEEE Trans. Knowl. Data Eng.* 25 (1) (2013) 220–232.

- [5] J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang, A. Steinfeld, Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design, in: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2011, pp. 1677–1686.
- [6] A. Noulas, S. Scellato, N. Lathia, C. Mascolo, Mining user mobility features for next place prediction in location-based services, in: *Data mining (ICDM), 2012 IEEE 12th international conference on*, IEEE, 2012, pp. 1038–1043.
- [7] S. Gambs, M.-O. Killijian, M. N. del Prado Cortez, Next place prediction using mobility markov chains, in: *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, ACM, 2012, p. 3.
- [8] W. Mathew, R. Raposo, B. Martins, Predicting future locations with hidden Markov models, in: *Proceedings of the 2012 ACM conference on ubiquitous computing*, ACM, 2012, pp. 911–918.
- [9] J. Krumm, E. Horvitz, Predestination: Inferring destinations from partial trajectories, in: *International Conference on Ubiquitous Computing*, Springer, 2006, pp. 243–260.
- [10] J. Krumm, E. Horvitz, Predestination: Where do you want to go today?, *Computer* 40 (4).
- [11] E. Horvitz, J. Krumm, Some help on the way: Opportunistic routing under uncertainty, in: *Proceedings of the 2012 ACM conference on Ubiquitous Computing*, ACM, 2012, pp. 371–380.
- [12] A. Y. Xue, J. Qi, X. Xie, R. Zhang, J. Huang, Y. Li, Solving the data sparsity problem in destination prediction, *The VLDB Journal/The International Journal on Very Large Data Bases* 24 (2) (2015) 219–243.
- [13] B. D. Ziebart, A. L. Maas, A. K. Dey, J. A. Bagnell, Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior, in: *Proceedings of the 10th international conference on Ubiquitous computing*, ACM, 2008, pp. 322–331.
- [14] G. Wang, J. Kim, The Prediction of Traffic Congestion and Incident on Urban Road Networks Using Naive Bayes Classifier, in: *Australasian Transport Research Forum (ATRF)*, 38th, 2016, Melbourne, Victoria, Australia, 2016.
- [15] J. D. Mazimpaka, S. Timpf, Trajectory data mining: A review of methods and applications, *Journal of Spatial Information Science* 2016 (13) (2016) 61–99.
- [16] S. Choi, H. Yeo, J. Kim, Network-Wide Vehicle Trajectory Prediction in Urban Traffic Networks using Deep Learning, *Transportation Research Record* (2018) 0361198118794735.
- [17] J. L. Adler, Investigating the learning effects of route guidance and traffic advisories on route choice behavior, *Transportation Research Part C: Emerging Technologies* 9 (1) (2001) 1–14.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, \. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
- [19] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: *International conference on machine learning*, 2015, pp. 2048–2057.
- [20] T. Cabannes, M. A. S. Vincentelli, A. Sundt, H. Signargout, E. Porter, V. Fighiera, J. Ugirumurera, A. M. Bayen, The impact of GPS-enabled shortest path routing on mobility: a game theoretic approach 2, *Tech. rep.* (2018).
- [21] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (8) (1997) 1735–1780.
- [22] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.
- [23] J. Kim, K. Zheng, S. Ahn, M. Papamanolis, P. Chao, Graph-based analysis of city-wide traffic dynamics using time-evolving graphs of trajectory data, in: *Australasian Transport Research Forum (ATRF)*, 38th, 2016.
- [24] J. Kim, K. Zheng, J. Corcoran, S. Ahn, M. Papamanolis, Trajectory Flow Map: Graph-based Approach to Analysing Temporal Evolution of Aggregated Traffic Flows in Large-scale Urban Networks, *Tech. rep.* (2017).
- [25] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: *Proceedings of the 40th annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2002, pp. 311–318.
- [26] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.