



Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks

Jie Sun, Jiwon Kim *

School of Civil Engineering, The University of Queensland, St. Lucia 4072, Brisbane, Australia

ARTICLE INFO

Keywords:

Vehicle trajectory
Next location prediction
Travel time prediction
Deep learning
LSTM
Self-attention

ABSTRACT

This paper aims to incorporate travel time prediction in the next location prediction problem to enable the prediction of the city-wide movement trajectory of an individual vehicle by considering both where the vehicle will go next and when it will arrive. We propose two deep learning models based on long short-term memory (LSTM) neural networks with self-attention mechanism—namely, hybrid LSTM and sequential LSTM. These models capture patterns in location and time sequences in trajectory data and their dependencies to predict next locations and travel times simultaneously. Using Bluetooth vehicle trajectory data from Brisbane, Australia, we compare the prediction performance of the proposed models with several existing approaches including hidden Markov model and other LSTM-based models. The results show that the proposed models produce higher prediction accuracy for both location and time prediction tasks, with the sequential LSTM yielding the best performance. Compared to the conventional next location prediction problem, which considers location sequences only without travel time consideration, the study finds that jointly modelling location and travel time sequences actually improves the next location prediction performance itself, potentially because travel time observations capture the information on traffic conditions in the network, which may affect drivers' location choices. We demonstrate an application of the proposed models in network traffic management, where important locations can be identified to mitigate congestion in a hot-spot by predicting where vehicles come from and go next in an urban road network.

1. Introduction

The increasing use of location-aware sensors such as GPS, WiFi, and Bluetooth devices has produced large amounts of urban trajectory data, which capture the spatio-temporal footprint of the movement of individual people and vehicles travelling around cities. Trajectory data contain rich information on travel behaviours and network-wide traffic dynamics and offer new opportunities to discover and predict individual and collective mobility patterns of users in urban road networks (Kim and Mahmassani, 2015; Chen et al., 2020b). Individual mobility prediction is of great importance for proactive traffic management and many location-based services (LBS) in intelligent transportation systems (Kulkarni and Garbinato, 2019). Individual mobility prediction aims to forecast the spatio-temporal movement patterns of an individual user based on the user's historical mobility records. Specific questions that are

* Corresponding author.

E-mail addresses: jie.sun@uq.edu.au (J. Sun), jiwon.kim@uq.edu.au (J. Kim).

commonly addressed include next location prediction, destination prediction, point-of-interest (POI) prediction, and so on. Such knowledge enables traffic engineers to predict how traffic flow propagates in space and time across the network and deploy traffic control or route guidance strategies to alleviate traffic congestion (Liu et al., 2019). The LBS providers and emerging Mobility-as-a-Service (MaaS) industry can also benefit from individual mobility prediction by identifying high demand areas and providing better services to meet users' personalised trip desire.

The next location prediction problem aims to predict the next location in a trajectory of a user based on the previous locations in the trajectory. There has been a considerable amount of work on the subject of predicting a user's next location, which is reviewed in detail in the literature review section below. Most of the existing studies, however, mainly focus on the spatial aspect of a user's movement (e.g., travel route) and do not consider the temporal aspect (e.g., travel time, arrival time, and duration). To better understand the dynamics of vehicular flow and transportation system performance, it is desirable to predict the travel route and travel time simultaneously. For instance, by predicting where and when road users will go next, traffic managers and operators could design active management and control strategies to ease potential congestion in advance. Incorporating travel time prediction in the next location prediction problem would also be beneficial from a modelling perspective as a driver's route choice tends to be affected by the travel times experienced by the driver and, thus, capturing the dependency between location and time sequences could improve the location prediction accuracy.

With the increasing availability of trajectory data and advances in deep learning, there has been a growing interest in applying deep learning to learn sequential patterns in trajectory data for the purpose of predicting future trajectories. Different types of trajectory prediction problems have been addressed across a broad range of application areas, including predicting city-wide activity location trajectories throughout the day for human mobility analysis (Krishna et al., 2018; Chen et al., 2020b), predicting pedestrian trajectories with a prediction horizon of several minutes (e.g., 15 min) for crowd management (Duives et al., 2019), and predicting pedestrian or vehicle movements with a very short prediction horizon (e.g., 0.1–1 s) for trajectory planning of connected and/or

Table 1
Representative studies of location and time prediction.

Study	Purpose	Method	Dataset
Ashbrook and Starner (2002, 2003)	Location prediction	Markov model	GPS data
Mathew et al. (2012)	Location prediction	Hidden Markov model	GeoLife dataset
Gambs et al. (2012)	Location prediction	Mobility Markov chain	Phonetic dataset, GeoLife dataset and synthetic dataset
Amirrudin et al. (2013)	Location prediction	Markov chain	Log file
Lu et al. (2013)	Location prediction	Markov chain based models	Call detail records of mobile phone
Baratchi et al. (2014)	Location prediction	Hierarchical hidden semi-Markov-based model	Geolife dataset, Capricorn dataset
Rathore et al. (2019)	Location prediction	A scalable clustering and Markov chain-based hybrid framework	Taxi trajectory dataset
Liu et al. (2019)	Location prediction	Travel time difference and Markov model	Vehicle passage records and taxi trajectory
Pathirana et al. (2003)	Location prediction	Extended Kalman Filter	GPS and acceleration of a mobile-user
Yavaş et al. (2005)	Location prediction	Mobility rules	Generated user paths
Monreale et al. (2009)	Location prediction	Decision tree	GeoPKDD project dataset
Noulas et al. (2012)	Location prediction	Linear regression and M5 model tree	Foursquare check-in data
Ying et al. (2014)	Location prediction	Geographic-temporal-semantic pattern	EveryTrail dataset, Bikely dataset
Alhasoun et al. (2017)	Location prediction	Dynamic Bayesian network	Call detail records of mobile phone
Chen et al. (2019)	Location prediction	Mobility pattern embedding model	Vehicle passage records and taxi trajectory
Liu et al. (2016)	Location prediction	RNN	Global Terrorism Database and Gowalla dataset
Endo et al. (2017)	Location prediction	RNN	Taxi trajectory dataset and GeoLife dataset
Choi et al. (2018)	Location prediction	RNN	Bluetooth data
Choi et al. (2019)	Location prediction	Attention based LSTM	Bluetooth data
Zhang et al. (2019)	Location prediction	CNN and LSTM	GeoLife dataset and Ningbo AIS data
Li et al. (2020)	Location prediction	Fuzzy LSTM	Mobile communication signalling dataset
Chen et al. (2020a)	Location prediction	Convolutional embedding model	Vehicle passage records and taxi trajectory
Scellato et al. (2011)	Location and time prediction	Nonlinear time series analysis	GPS data and WIFI data
Gidófalvi and Dong (2012)	Location and time prediction	Dynamically weighted ensemble of Inhomogeneous Continuous-Time Markov models	GeoLife dataset
Do and Gatica-Perez (2012)	Location and time prediction	A probabilistic framework	Nokia mobile phone data
Du et al. (2016)	Location and time prediction	RNN and temporal point process	New York City Taxi dataset
Krishna et al. (2018)	Activity and time prediction	Hybrid and cascaded LSTM	Sensor data containing the activities and durations
Chen et al. (2020b)	Location and time prediction	Context-aware deep model called DeepJMT	Foursquare check-in data

autonomous vehicles (Huang et al., 2018; Jain et al., 2020; Ivanovic et al., 2020; Wang et al., 2020). There has been, however, little research done in the context of network traffic flows in urban road networks, where the goal is to predict the locations and arrival times of vehicle trajectories for the purpose of real-time traffic management (with a typical prediction horizon ranging from 5 min to 1 h). This study aims to fill this gap and develop data-driven models that can learn the interaction between drivers' dynamic route choice behaviours and en-route travel time experiences from large-scale trajectory data to predict individual vehicles' complete trajectories to better understand network traffic flows.

More specifically, we aim to learn location and travel time patterns from network-wide urban vehicle trajectory data, where each trajectory is represented as a sequence of locations (e.g., road segments or intersections) and timestamps along a travel path of an individual vehicle. Since our task is to learn patterns in sequential data, we choose to use Long Short-Term Memory neural networks (LSTM), which have been known for its superiority in learning the temporal dependency in sequential data (Hochreiter and Schmidhuber, 1997). In particular, we propose two deep learning models based on LSTM—namely, hybrid LSTM and sequential LSTM—to perform location and time prediction tasks jointly. The hybrid LSTM model predicts the location and travel time outputs with the same inputs and hidden layers by sharing parameters between the two tasks, whereas the sequential LSTM model conducts the two tasks sequentially with two closely linked modules, each of which has its own parameter set. While LSTM can pass relevant information down the long chain of input sequences to make predictions, it has some limitation in relating different positions within the same input sequence. To address this limitation, we incorporate a self-attention mechanism (Zheng et al., 2018; Vaswani et al., 2017) into our LSTM models, which helps capture the correlation between different positions in a trajectory and enhance the performance of the LSTM models. With the Bluetooth data in Brisbane downtown area, we train and evaluate the proposed models and compare them with baseline models (basic LSTM models and hidden Markov model for location prediction, and historical average model and LSTM model for travel time prediction). In addition, we investigate the impact of spatial resolution of trajectory data on the prediction accuracy by testing models with four different datasets with different Bluetooth detector density. We demonstrate an application of the proposed models in network traffic flow analysis, where the predicted vehicle trajectories can be used to estimate the connectivity between locations to identify road segments that carry major traffic demand to a particular target location or congestion hot-spot and determine where to control to prevent potential excessive demand in the target location.

The remainder of this paper is organized as follows: Section 2 reviews the literature on next location and time prediction; Section 3 introduces the proposed hybrid and sequential LSTM models; Section 4 presents the study area and data preparation; Section 5 evaluates the proposed models with real-world trajectory data; Section 6 demonstrates an application of the proposed models, and Section 7 summarizes and discusses the main findings of this study.

2. Literature review

Over the last two decades, many efforts have been made in terms of predicting the next location that a user will visit in the future based on urban trajectory data. A number of studies also attempted to predict the time that the user will arrive at the next location (or the duration that the user will stay at the next location). In this section, we review recent studies addressing next location prediction problems from two perspectives, namely *location prediction* and *location and time prediction*. A summary of the reviewed studies is presented in Table 1.

2.1. Location prediction

Location prediction focuses on the future locations a user will visit by capturing the underlying travel patterns from historical data. Ashbrook and Starner (2002, 2003) are two of the first influential works that predict user location with GPS information. They clustered the GPS data to extract significant locations and then developed a Markov model to predict user's movement for both single- and multi-user scenarios. Many approaches along this line have been presented using location datasets in the last two decades. Mathew et al. (2012) adopted a hidden Markov model to predict pedestrian movement using the GeoLife dataset. Gambs et al. (2012) used a mobility model called Mobility Markov Chain to predict the next location of an individual. Amirrudin et al. (2013) predicted user's movement via Markov chains with input of user's mobility history. Lu et al. (2013) implemented a series of Markov chain-based models to predict the actual locations visited by each user. Baratchi et al. (2014) proposed a hierarchical hidden semi-Markov model to understand the behavior of mobile entities. Rathore et al. (2019) developed a scalable clustering and Markov chain-based hybrid framework for both short- and long-term location prediction. Liu et al. (2019) integrated the difference between the shortest travel time and the actual travel time with a Markov model to predict next locations.

Besides Markov-based models, other statistical and machine learning models have been used to predict locations in mobility trajectories. Pathirana et al. (2003) provided a scheme for mobility estimation and used robust extended Kalman Filter to predict mobile user's next location. Yavaş et al. (2005) mined user mobility patterns from historical trajectories and extracted mobility rules from these patterns for mobility predictions. Monreale et al. (2009) extracted trajectory patterns of moving objects with a decision tree and used it as a predictor of the next location of a new trajectory. Noulas et al. (2012) proposed a set of features, including transitions between types of places, mobility flows between venues, and spatial-temporal characteristics of user check-in patterns, to capture the factors that may drive users' movements and predict users' next places based on linear regression and M5 model tree. Ying et al. (2014) considered a user's geographic-triggered intentions, temporal-triggered intentions, and semantic-triggered intentions simultaneously to capture the trajectory pattern and estimate the probability that the user visits a certain location. Alhasoun et al. (2017) investigated the temporal and spatial similarity of mobility patterns of users and developed a dynamic Bayesian network to incorporate the information of other users with similar mobility characteristics to improve the accuracy of next location prediction for a given user. Chen

[et al. \(2019\)](#) proposed a mobility pattern embedding model to extract people's mobility patterns in traffic trajectory data from multiple aspects including sequential, personal, and temporal factors for next location prediction.

Recently, the advances in deep learning have facilitated next location prediction research with the enhanced ability to capture spatiotemporal dependencies in large-scale trajectory data. [Liu et al. \(2016\)](#) extended a Recurrent Neural Network (RNN) and proposed a method called Spatial Temporal RNN to model local temporal and spatial contexts in each layer and predict next location. [Endo et al. \(2017\)](#) employed RNN to predict a user's destinations from his or her partial movement trajectories. [Choi et al. \(2018\)](#) applied RNN to predict the next locations in a vehicle's trajectory by partitioning the network into cells. [Choi et al. \(2019\)](#) further incorporated an Attention mechanism into an RNN model to leverage the information on network traffic states to improve the prediction of individual vehicle movements. [Zhang et al. \(2019\)](#) proposed a Multi-task learning framework for location prediction by using Convolutional Neural Network (CNN) to extract spatial features and using Long Short-Term Memory (LSTM) network to extract the sequence and time attributes between the locations of moving objects. [Li et al. \(2020\)](#) introduced a fuzzy trajectory concept and extended a standard LSTM model to a fuzzy-LSTM for trajectory prediction. [Chen et al. \(2020a\)](#) proposed a convolutional embedding model to predict next locations using traffic trajectory data, by modelling the relative ordering of locations with a one-dimensional convolution.

2.2. Location and time prediction

In addition to predicting next locations, the ability to predict when an individual user will arrive at the next location (or the duration that the user will stay at the next location) can greatly enhance and expand a model's applicability in real-world traffic management and LBS solutions. While still limited, attempts to predict not only the next location but also the time or duration have grown in recent years. [Scellato et al. \(2011\)](#) presented an approach to location prediction based on non-linear time series analysis of the arrival and residence times of users in relevant places. [Gidófalvi and Dong \(2012\)](#) used a dynamically weighted ensemble of Inhomogeneous Continuous-Time Markov models to capture the temporal, periodic, and sequential regularities in movements of a user to predict when and where the user will move next. [Do and Gatica-Perez \(2012\)](#) proposed a probabilistic framework by extracting and combining different mobility patterns based on an ensemble method to predict next place and stay duration. [Du et al. \(2016\)](#) combined an RNN and a temporal point process as Recurrent Marked Temporal Point Process (RMTPP) to simultaneously model a user's visiting time and location. [Krishna et al. \(2018\)](#) developed two distinct LSTM networks to estimate the probabilities for future activities and their durations based on a sequence of past activities and durations. [Chen et al. \(2020b\)](#) combined a hierarchical RNN-based sequential dependency encoder, a spatial context extractor, a periodicity context extractor, and a co-attention-based social and temporal context extractor for joint mobility and time prediction.

In summary, the literature shows that location prediction for individual mobility has been widely studied while research on the simultaneous prediction of location and time is limited. In particular, predicting the next location and the travel time to the next location in the context of vehicular traffic in road networks has not yet been explored. The goal of this paper is to fill this gap by developing models that predict the spatial-temporal movement of an individual vehicle across a road network, with potential applications in traffic management.

3. Methodology

In order to learn and predict the next locations and travel times in urban trajectories simultaneously, we propose two methods of multi-prediction modelling based on LSTM networks: hybrid LSTM model and sequential LSTM model. We first give a brief overview of

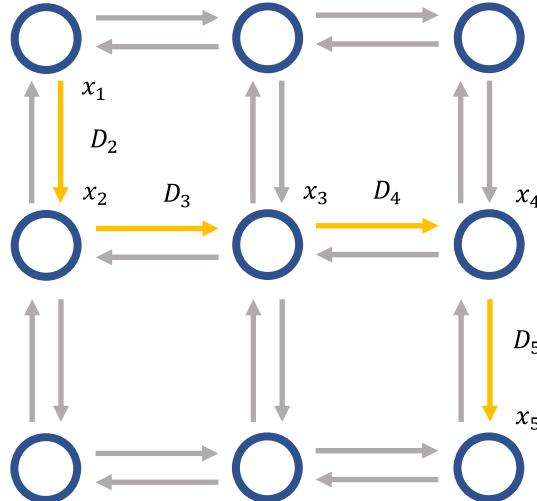


Fig. 1. Representation of network and trajectory.

a basic LSTM model and self-attention mechanism and then introduce our proposed methods.

3.1. Vehicle trajectories

Let $Tr = \{p_1, p_2, \dots, p_n\}$ represent a urban vehicle trajectory consisting of n_{data} points, where data point p_i represents the vehicle's i th position that consists of the information on its location x_i (latitude and longitude coordinate) and timestamp τ_i . Let D_i denote the travel time from the $(i-1)$ th location to the i th location, which can be obtained from $\tau_i - \tau_{i-1}$. Fig. 1 illustrates an example of this trajectory representation, where a vehicle traverses five locations from x_1 to x_5 resulting in four travel time records D_2 through D_5 . When trajectory data are supplied to the prediction models for training and testing, one special token, *end_token*, is added to the end of each trajectory's location sequence to indicate the end of trip.

3.2. The basic LSTM model

LSTM networks are a variant of RNN that can learn the temporal dependency in sequential data (Hochreiter and Schmidhuber, 1997). As LSTM can automatically extract and pass relevant information down the long chain of sequences to make predictions, it is suitable for learning sequential movement patterns in urban trajectory data and, thus, selected as the basis for our proposed framework to predict next locations and travel times. While LSTM was developed to address the well-known *vanishing* (or *exploding*) gradient problem in RNNs, LSTM does not completely eliminate this problem and may still have an issue with training very long sequences. However, this is not a critical concern in our models because this issue becomes critical when dealing with very long sequences (e.g., with more than 1000 time steps) and a sequence length less than 100 is considered short in this context (Li et al., 2018; Gers et al., 2002). In our study, trajectories usually have a length between 5 and 25, which are quite short, and thus our models do not suffer from the gradient vanishing and exploding problem. Even if we consider a whole city-wide network, it is unlikely that a single vehicle trajectory would contain more than 100 detection points in one trip—as our next location prediction problem considers link-to-link or intersection-to-intersection movement sequences, rather than microscopic second-by-second vehicle movements—and, thus, our models could still be used for a larger network without suffering from the vanishing/exploding gradient problem.

The structure of a standard LSTM model is shown in Fig. 2, which contains one input layer, one hidden layer, and one output layer. The input layer initializes input data for subsequent layers, while the output layer performs the final prediction task (classification or regression). The hidden layer is designed for learning features in input sequence through a recurrent unit called memory block. The memory block in the hidden layer contains a cell, which transfers relevant information throughout the sequence, and three gates, namely, forget gate, input gate, and output gate, which regulate information flow. The first step in an LSTM memory block is to decide what information to keep or forget from the output of the previous memory block, h_{t-1} , and the input of the current memory block, X_t . Then, the input gate captures the information from input data X_t and updates the cell state. Finally, the output gate decides which part of the cell state is used to compute the final output, h_t . The compact form of the equations is shown as below.

Forget gate:

$$f_t = \sigma(W_{fx}X_t + W_{fh}h_{t-1} + b_f) \quad (1)$$

Input gate:

$$i_t = \sigma(W_{ix}X_t + W_{ih}h_{t-1} + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_{Cx}X_t + W_{Ch}h_{t-1} + b_c) \quad (3)$$

Cell state update:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

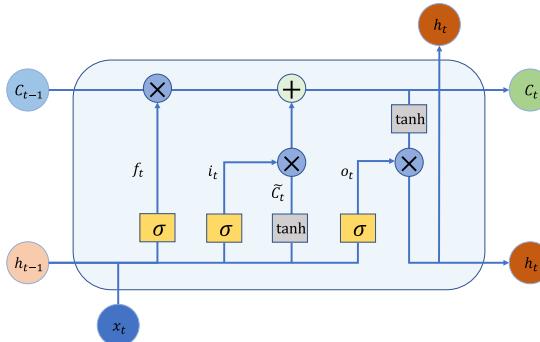


Fig. 2. Schematic diagram of the memory block of LSTM.

Output gate:

$$o_t = \sigma(W_{ox}X_t + W_{oh}h_{t-1} + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

where X_t is the input data at time step t and C_t is the hidden cell state at t . Variables f, i, C and o represent forget gate, input gate, cell state vectors, and output gate, respectively. Operator $*$ represents the scalar product of two vectors, $\sigma()$ is the standard logistic sigmoid function, which changes the inputs into range of $(0, 1)$, and $\tanh()$ is the hyperbolic tangent function, which transforms the input and output into range of $[-1, 1]$. Parameters $W_j (j = fx, fh, ix, ih, ox, oh)$ denote the weight matrices in each layer and $b_j (j = f, i, \tilde{C}, o)$ denotes the bias of each gate.

3.3. Self-attention

Accurately learning temporal dependencies in sequential data is a key challenge in many sequence prediction tasks. Although LSTM models can deal with long-sequence data and pass information across many time steps, it has limited ability to relate different positions within the input sequence. Specifically, in LSTM, information from input sequence X_1, \dots, X_t is aggregated into the current hidden state h_t on a position-by-position basis in sequential order and there is no explicit mechanism for modeling relations between positions (Cheng et al., 2016). The concept of attention mechanism has been, thus, introduced to learn dependencies between distant positions in sequence data: either positions from two different sequences, namely *inter-attention* (Bahdanau et al., 2014) or positions within a single sequence, namely *intra-attention* or *self-attention* (Cheng et al., 2016; Vaswani et al., 2017). We adopt the self-attention mechanism in conjunction with the LSTM model to better capture the dependencies between locations within each trajectory sequence.

The self-attention mechanism allows the model to learn which previous input positions it should pay more *attention* to at each step of the out prediction. Fig. 3 shows an illustrative example of a trajectory with four locations, where the degree of dependencies between the next location (x_4) and the previous locations (x_1, x_2, x_3) is indicated by arrow width (thicker line indicates stronger dependency). It illustrates that, in predicting the next location (x_4) in the trajectory, the last visited location (x_3) is more informative, thus should be paid more attention to, than the previous location (x_2), but the origin point (x_1), although more distant, is even more informative than the location just visited. Capturing such varying degrees of dependencies across different locations in the network would be important in improving the prediction accuracy for our next location and travel time prediction problem.

The self-attention mechanism used in this study adopts the approach in Zheng et al. (2018), which is implemented as follows:

$$g_{t,t'} = \tanh(W_g h_t + W_{g'} h_{t'} + b_g) \quad (7)$$

$$e_{t,t'} = \sigma(W_a g_{t,t'} + b_a) \quad (8)$$

$$a_{t,t'} = \frac{\exp(e_{t,t'})}{\sum_j \exp(e_{t,j})} \quad (9)$$

$$A_t = \sum_{t'} a_{t,t'} h_{t'} \quad (10)$$

where h_t and $h_{t'}$ are the hidden state representations at current time step t and previous time step t' from the LSTM layer; σ is the element-wise sigmoid function; W_g and $W_{g'}$ are the weight matrices corresponding to h_t and $h_{t'}$; W_a is the weight matrix corresponding to their non-linear combination; and b_g and b_a are the bias vectors. The attention output at time step t , A_t , is the weighted summation of all previous hidden states $h_{t'}$, weighted by $a_{t,t'}$, where $a_{t,t'}$ captures the similarity or dependency between h_t and $h_{t'}$ (i.e., the relation between the current position at t and the previous position at t' in a given input trajectory).

3.4. The proposed models

Having introduced the structure of basic LSTM model and self-attention mechanism, we present the structure of the proposed hybrid and sequential LSTM model in this section.

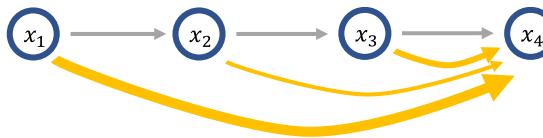


Fig. 3. Dependency between different locations in a trajectory.

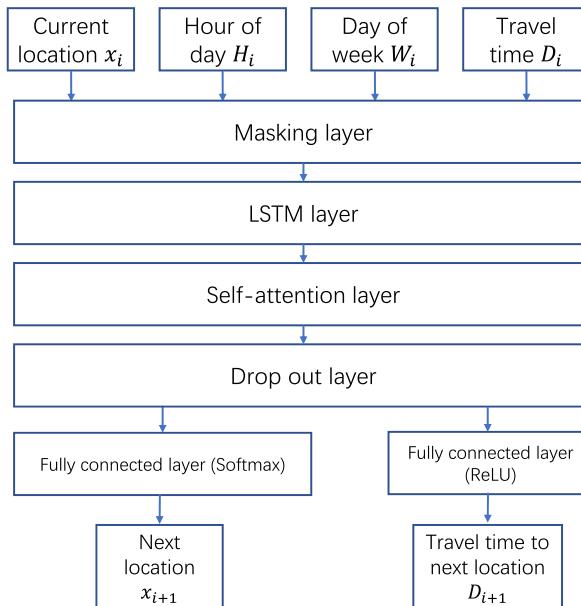
3.4.1. Hybrid LSTM model

The hybrid LSTM model predicts both the next location and the travel time between the current and next locations by attempting to simultaneously minimize the corresponding location and time prediction errors within a single LSTM network. The architecture of hybrid LSTM model is illustrated in Fig. 4 and described below. We use index i to represent the position in a trajectory.

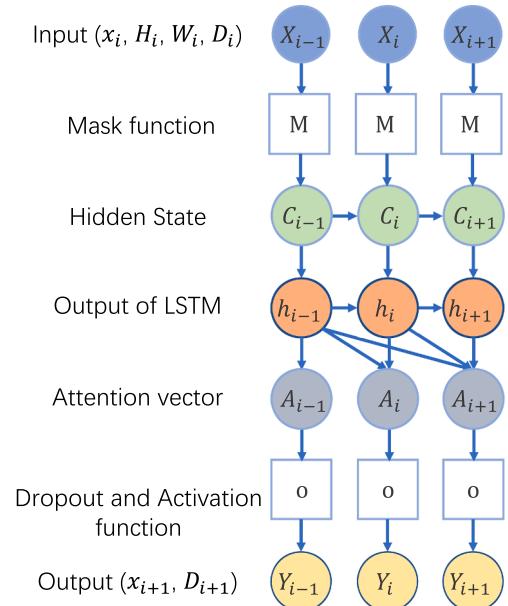
- Input Layer: The hybrid LSTM model takes four data sequences as input: the current location (x_i) that a vehicle visits, which is encoded as a one-hot vector; the hour of day (H_i) and the day of week (W_i) of the trip which is in the form of a one-hot vector of size 24 and 7, respectively; the travel time from the previous location to the current location (D_i), which is normalised to [0, 1]. The input sequences are concatenated and transferred to the next layer.
- Masking Layer: The encoded input then goes through a masking layer which can mask a sequence by using a mask value to skip time steps to deal with the variable-length input, as the length of different vehicles' trip are different.
- LSTM Layer: The masked input is fed into an LSTM layer, as described in Section 3.2, which outputs the hidden state representation h_i (as in Eq. 6) that stores historical information and is carried forward to the subsequent time steps.
- Self-attention Layer: The attention vector A_i (as in Eq. 10) is then calculated based on all the previous hidden states from the LSTM layer through the self-attention layer, as described in Section 3.3.
- Dropout Layer: Dropout is an effective technique that randomly ignores neurons during training to prevent the over-fitting of neural networks (Srivastava et al., 2014).
- Fully-connected Layer (also known as activated dense layer): The dense layer computes a vector which represents (i) the probabilities of next locations with a softmax activation function for location prediction or (ii) the expected normalised travel time from the current location to the next location with a rectified linear unit (ReLU) activation function for travel time prediction.
- Output Layer: The layer combines and produces the two outputs from the last layer, i.e., probability of next locations and travel time at each time step.

3.4.2. Sequential LSTM model

Instead of building a single LSTM network for both location and time prediction tasks as in the hybrid LSTM model, the sequential LSTM model has a separate LSTM network for each task and connects these two LSTM networks sequentially such that the next location is predicted first and then the associated travel time is predicted. Compared to the hybrid LSTM model that shares the same set of parameters between the two tasks, the sequential LSTM model maintains separate parameter sets, providing more flexibility in learning the task-specific features. The architecture of sequential LSTM model is shown in Fig. 5. In the location prediction LSTM (left), the current location (x_i), the hour of day (H_i), and the day of week (W_i) of the trip are the inputs to the masking, LSTM, self-attention, dropout, and dense layers to predict the next location (x_{i+1}). The time prediction LSTM (right) then combines the attention vector from the first LSTM network and the travel time from the last location to the current location (D_i) to predict the next travel time (D_{i+1}) using another set of layers. It is noted that although location prediction and travel time prediction are done through separate LSTM modules



(a) Layers of hybrid LSTM model



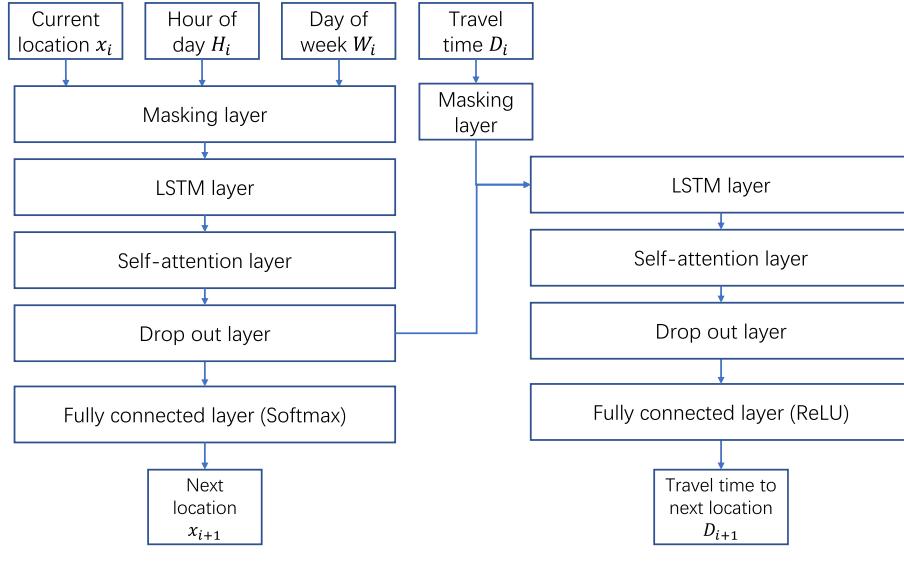
(b) Data flow of hybrid LSTM model

Fig. 4. The architecture of hybrid LSTM model.

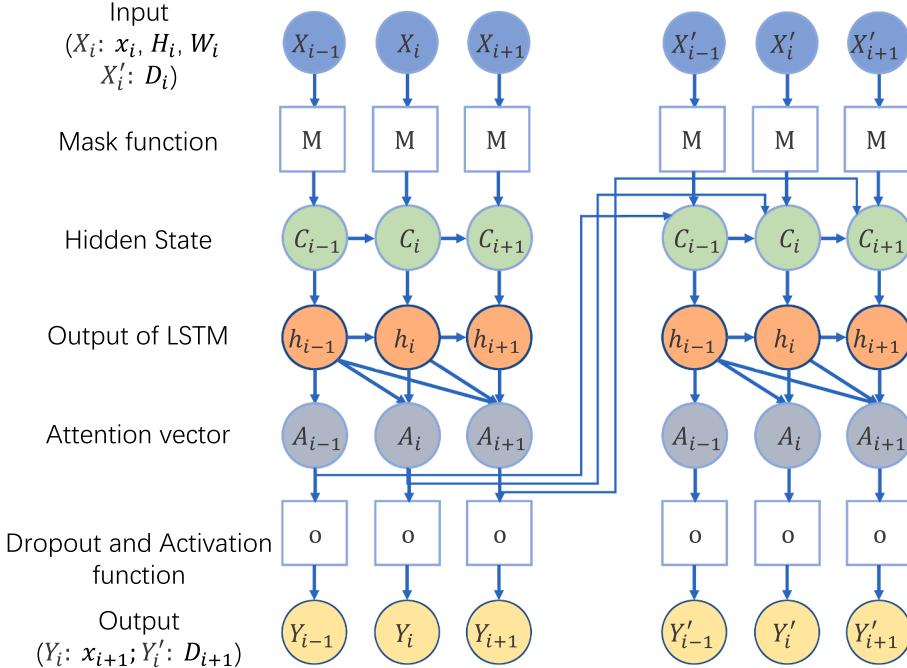
in the sequential LSTM model, these two tasks still influence each other as the model parameters are updated based on the performance on both tasks. For instance, the performance on travel time prediction influences the parameters of the location prediction module through the backpropagation during training, while the performance on location prediction influences the parameters of the travel time prediction as the information used to predict the next location is also passed to the travel time prediction layers.

3.5. Model configuration

With the proposed structures of LSTM models, several configurations of LSTM need to be specified to optimize the proposed model,



(a) Layers of sequential LSTM model



(b) Data flow of sequential LSTM model

Fig. 5. The architecture of sequential LSTM model.

such as the loss function, activation function, and optimization algorithm.

3.5.1. Loss function

The loss function, also known as objective function, which calculates the error (the loss) between the predicted and the observed values, is used to find the optimal parameters minimising the loss function. Loss functions in machine learning can be broadly divided into classification loss for discrete variables and regression loss for continuous variables. Given the classification (location prediction) and regression (travel time) problems in this study, we adopt categorical cross-entropy for the location prediction and mean absolute error (MAE) for the travel time prediction to describe the loss associated with each trajectory as shown below:

$$L(Y, \hat{Y}) = \sum_{i=2}^n -\frac{1}{|J|} \sum_{j \in J} \left(Y_{ij} \log \hat{Y}_{ij} + (1 - Y_{ij}) \log (1 - \hat{Y}_{ij}) \right) \quad (11)$$

$$L(D, \hat{D}) = \frac{\sum_{i=2}^n |D_i - \hat{D}_i|}{n-1} \quad (12)$$

where $L(Y, \hat{Y})$ is the loss between the observed location sequence (Y) and the predicted location sequence (\hat{Y}) in a given trajectory, $L(D, \hat{D})$ is the loss between the observed travel time sequence (D) and the predicted travel time sequence (\hat{D}), n is the trajectory length, J is the set of detector locations, Y_{ij} is the binary label set to 1 if location j is visited at the $(i+1)$ th step in the sequence or 0 otherwise, and \hat{Y}_{ij} is the predicted probability that location j is visited at the $(i+1)$ th step; and D_i and \hat{D}_i are the real and expected travel time from location $i-1$ to location i , respectively.

To effectively minimise the two loss functions at the same time, we combine them with a weight for the MAE. Therefore, the complete loss function is shown as:

$$L = L(Y, \hat{Y}) + wL(D, \hat{D}) \quad (13)$$

where w is the weight for the MAE loss function, which is set as 8 in this study based on our experiments for hyperparameter tuning.

3.5.2. Activation function

The activation function adds a non-linear transformation to the inputs of the neurons, which can improve the ability of nonlinear modeling (Hochreiter and Schmidhuber, 1997). The softmax function and Rectified Linear Unit (ReLU) function are used as the



Fig. 6. Bluetooth detectors in the study area of Brisbane.

activation function for location and travel time prediction, respectively.

3.5.3. Optimization algorithm

For the optimization algorithm to perform stochastic gradient descent to minimize the loss functions, the Root Mean Square Propagation (RMSprop) method, which is one of the most popular and fast optimization algorithms, is used in this study ([Tieleman and](#)

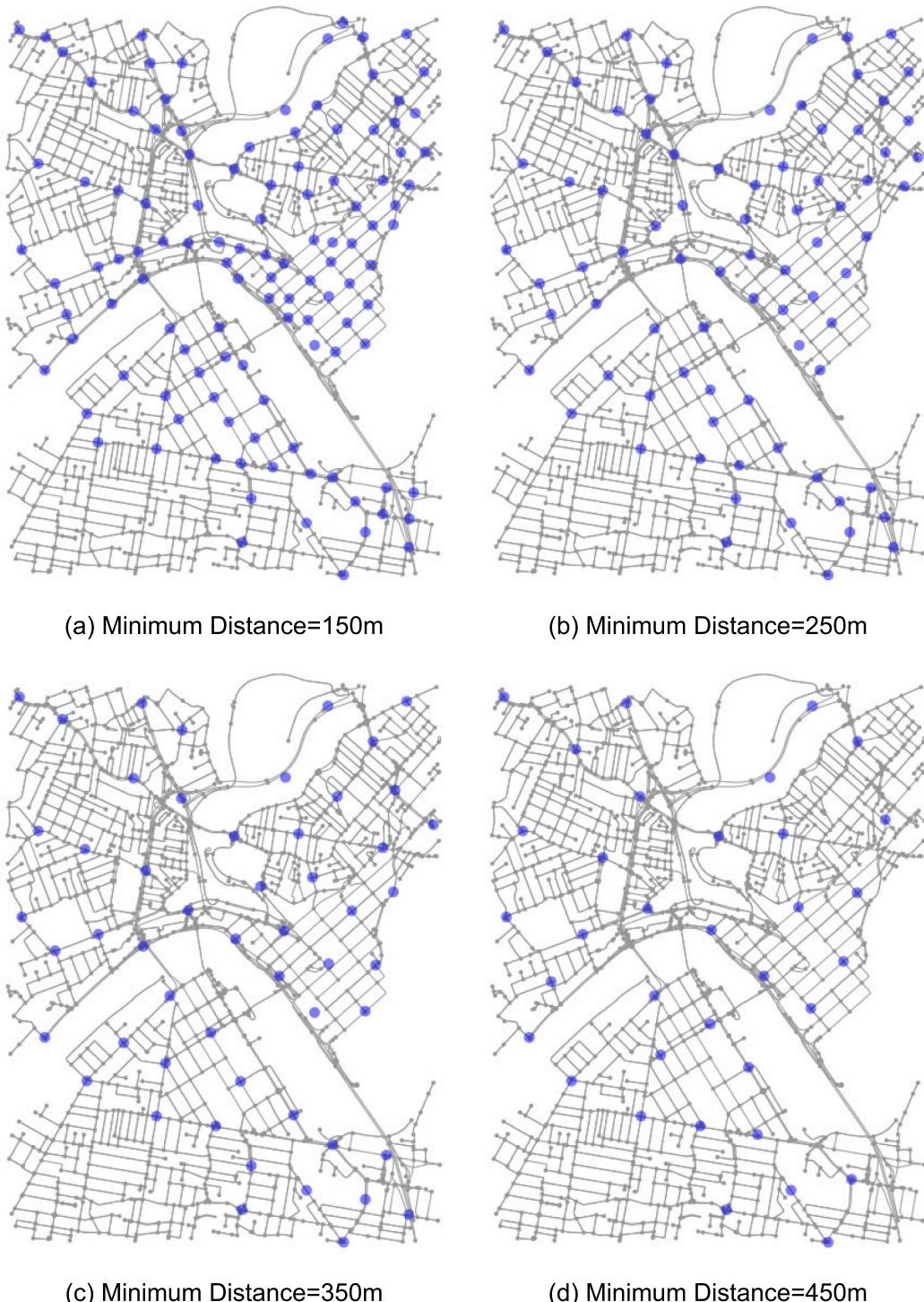


Fig. 7. Bluetooth detectors in sub-networks with different detector densities.

Hinton, 2012).

4. Data preparation

4.1. Data and study area

Recently, Bluetooth data have emerged as a new source of information to understand vehicle movements in urban road networks (Bhaskar and Chung, 2013; Laharote et al., 2015; Michau et al., 2017). Bluetooth data collection is performed by installing stationary roadside Bluetooth detectors across a network and tracking the Media Access Control (MAC) addresses of individual (discoverable) Bluetooth devices (e.g., in-vehicle navigation systems and mobile devices) that cross the detection zones. In this study, we use the Bluetooth dataset collected from July 2015 to June 2016 in the downtown area in Brisbane, Australia, which were provided by the Department of Transport and Main Roads (TMR) and Brisbane City Council (BCC). The Brisbane Bluetooth dataset contains information on the identification of a Bluetooth detector, the identification of a Bluetooth device (device ID)—the encrypted, anonymised MAC address of that device—, and the time at which the device was detected. By tracking unique device IDs, the trajectories of individual vehicles can be constructed, where each trajectory represents a time-ordered sequence of Bluetooth detector locations that a given vehicle passed.

[Fig. 6](#) shows the study network with the Bluetooth detectors (blue dots) considered in this study. There are a total of 181 detectors in the study area, where the minimum distance between detectors is approximately 100 m. In addition to this original network, we also created networks with a subset of Bluetooth detectors to investigate the effect of spatial resolution (or detector density) on the model performance. We generated four different networks by randomly selecting a subset of detectors with the minimum distance between two detectors set to 150 m, 250 m, 350 m, and 450 m, respectively. The number of detectors in these sub-networks are 115, 79, 53, and 35, respectively, as shown in [Fig. 7](#). As the number of detectors decreases in a given study area, the distance between detectors increases and the resulting trajectories have lower spatial resolution (the distance between current and next locations is greater) and less data points (the sequence length is shorter). The purpose of this additional analysis is to understand how these factors—sensor network density and data characteristics—affect the accuracy of next location and travel time prediction.

For each of five detector networks (the original and four detector subsets), we process the Bluetooth data to construct the associated trajectory datasets. The summary information of these five trajectory datasets is provided in [Table 2](#).

4.2. Data processing

To construct accurate trajectories of vehicles, the Bluetooth data are cleaned and processed with the following steps.

- Filtering non-unique device IDs. The same device ID might be shared by different vehicles because of the possibility of cloning Bluetooth device parameters for fleet's specific needs (Michau et al., 2017). These shared IDs are excluded if they have anomalous speed more than ten times in five days (speed above 130 km/h are considered anomalous as the maximum speed limit in this network is 80 km/h).
- Integrating neighbouring detectors. As Bluetooth detectors scan device IDs over a certain area—usually a scanner with a 5 dBi omni directional antenna scans the device IDs over a circular radius of 100 m (Abedi et al., 2015)—, the detectors located in close distance have overlapping detection areas, which may lead to the wrong detection order of the vehicle (e.g., a downstream detector might scan a device before the upstream one does). We thus consider the detectors within 100 m as one detector group to avoid the detection order issue. Therefore, 181 detectors in the study area merge as 168 detectors (detector groups).
- Filtering duplicate scanners. A vehicle could be consecutively detected by a single scanner for several times. We thus only identify the first detection time and remove the others from the trajectories.
- Cutting detection sequences to trips. To study a consecutive trip from its origin to destination without parking, the detection sequence should be cut into several trips. As per Carpenter et al. (2012) and Michau et al. (2017), 30 min is selected as the threshold for splitting trips. In other words, it is assumed that a Bluetooth trajectory is split into different trips if the time gap between two consecutive points exceeds 30 min. After splitting trips, we discard very short trips to ensure that sufficient data points exist to learn sequential patterns when training models. The criteria we use is that a valid trip should be longer than or equal to 5 min and traverse at least 5 locations.

It noted that there are some limitations in the Bluetooth data used in this study even after the data are cleaned and pre-processed.

Table 2
Summary of trajectory data.

Dataset	Minimum distance between detectors (m)	No. of detectors	No. of trajectories
Original dataset	100	168	42,697,732
Sub-dataset 1	150	115	2,646,028
Sub-dataset 2	250	79	1,188,555
Sub-dataset 3	350	53	352,704
Sub-dataset 4	450	35	145,382

First, the Bluetooth data do not represent the whole vehicle population because not all the vehicles are equipped with Bluetooth devices and also not all the Bluetooth devices may be detected by the Bluetooth sensors. Second, there are possibilities that some sensor locations along a Bluetooth-equipped vehicle's path may be skipped due to missed detections, thereby producing trajectories that may not reflect the true next location information. Despite these limitations, the current dataset fulfills the intended purpose of this study as we focus on the methodology to learn 'observed' patterns in any given trajectory dataset, whether or not the observed patterns reflect the true underlying vehicle movements.

4.3. Training and testing dataset

A summary of extracted trip data for five different spatial resolutions (one original dataset and four sub-datasets) is presented in **Table 2**. To make the results of different datasets comparable, we use the same number of data for training, validation, and testing for all the cases. We randomly selected 100,000 vehicle trajectories for training, 10,000 vehicle trajectories for validation and 10,000 vehicle trajectories for testing for each dataset.

5. Results and Discussions

5.1. Performance measures

To evaluate the performance of the prediction, we adopt a series of indicators to compare the location and travel time prediction results of different models. In terms of location prediction, we employ BLEU (bilingual evaluation understudy) score (Papineni et al., 2002), which is one of the most popular metrics in sequence modelling (particularly language modelling), to evaluate the predicted trajectory. BLEU score uses a modified form of n -gram precision (n -gram is a contiguous sequence of n items from a given sequence) to compare how close two sequences are, as shown below:

$$p_n = \frac{\sum_{g \in C_n} \min[m_{\text{pred}}(g), m_{\text{obs}}(g)]}{\sum_{g \in C_n} m_{\text{pred}}(g)} \quad (14)$$

where p_n is the modified n -gram precision which represents the fraction of the n -grams (n -consecutive location blocks) in the predicted trajectory that are also found in the observed trajectory; C_n is the set of unique n -consecutive location blocks in the predicted trajectory; $m_{\text{pred}}(g)$ is the number of times a given n -consecutive location block, g , occurs in the predicted trajectory; and $m_{\text{obs}}(g)$ is the number of times n -consecutive location block g occurs in the observed trajectory. Once individual n -gram precisions, p_n , are calculated for different $n = 1, \dots, N$, the BLEU score of order N , denoted by BLEU_N , is then obtained by calculating the weighted geometric mean of these N individual n -gram precisions as follows:

$$\text{BLEU}_N = \min \left[1, e^{1 - \frac{l_{\text{obs}}}{l_{\text{pred}}}} \right] \exp \left(\sum_{n=1}^N w_n \log p_n \right) \quad (15)$$

where w_n is the weight for each n -gram precision (usually set to $w_n = 1/N$ for all n); l_{pred} and l_{obs} are the number of locations (sequence length) in the predicted and observed trajectories, respectively; and the term ' $\min[1, \exp(1 - l_{\text{obs}}/l_{\text{pred}})]$ ' is a multiplicative factor called *brevity penalty* that is introduced to penalize a short predicted trajectory (i.e., when l_{pred} is shorter than l_{obs}) because short trajectories can produce very high precision scores even if the prediction is poor. BLEU_N basically measures how well a model can predict n -consecutive locations in a given trajectory from $n = 1$ up to $n = N$. The better the location blocks in the predicted trajectory match those in the observed trajectory, the higher the BLEU score.

In addition, we use the accuracy (Acc_n) of predicting the first n consecutive locations, defined as the proportion of the number of trajectories which correctly predicted the first n consecutive locations to the total number of trajectories in the testing dataset, to assess the model performance (n is set as 1, 2, and 3 in this study). Both BLEU and Acc scores take a value between 0 and 1, with 0 representing a perfect mismatch and 1 representing a perfect match between the true and predicted sequences.

Regarding the travel time prediction, we adopt two widely used measures for regression analysis to evaluate the model performance, i.e., root mean square error (RMSE) and mean absolute percentage error (MAPE), as shown in Eqs. 16 and 17:

$$\text{RMSE} = \frac{1}{m} \sum_{j=1}^m \sqrt{\frac{1}{|K_j|} \sum_{k \in K_j} |D_k^* - D_k|^2} \quad (16)$$

$$\text{MAPE} = \frac{1}{m} \sum_{j=1}^m \frac{1}{|K_j|} \sum_{k \in K_j} \frac{|D_k^* - D_k|}{D_k} \quad (17)$$

where m is the number of trajectories, K_j is the set of correctly predicted next location pairs for trajectory j (i.e., ordered, consecutive

location pairs that are found in both the predicted and observed trajectories), and D_k^* and D_k are the predicted and observed travel time from location pair k , respectively.

5.2. Baseline models

Since there are no established models performing the location-time joint prediction task that can be used as baseline models for this study, we consider separate baseline models for location prediction and time prediction. Considering the wide use of Markov-based models in the next location prediction problem as discussed in the literature review, we use a hidden Markov model (HMM) as a baseline model for location prediction. HMM assumes that the observed sequence of locations are probabilistically generated as the output of a system with a hidden state, where the hidden state variable evolves like a Markov process governed by a transition probability matrix. The probability of observing a certain output sequence given a hidden state is described by a emission probability matrix (MacDonald and Zucchini, 1997). Therefore, the core of HMM is to estimate the transition matrix (between hidden states) and emission matrix (between hidden state and observed sequence) and generate an observed sequence based on the transition and emission matrices.

In addition to the HMM, we also consider deep learning models for location prediction. Two models tested are an LSTM model without an self-attention layer, referred to as LSTM-L (LSTM for location prediction only), and an LSTM model with a self-attention layer, referred to as Attention-LSTM-L (Attention-LSTM for location prediction only), respectively. The Attention-LSTM-L is the same as the proposed sequential LSTM model but without the travel time prediction component (i.e., the LSTM network on the left in Fig. 5). Although the location prediction part of the Attention-LSTM-L model and the proposed sequential LSTM model is the same in terms of modeling architecture and input data, the location prediction results from these two models are expected to be different because the travel time prediction component in the sequential LSTM model—although it is performed later than the location prediction—would affect the parameter update for the location prediction component through the backpropagation mechanism during the training process. More specifically, the sequential LSTM model performs location prediction and travel time prediction sequentially, but not independently, and the information from the loss function measuring the error in travel time prediction propagates backward to update not only the parameters of travel time prediction module, but also the parameters for the location prediction module. The comparison between Attention-LSTM-L and the sequential LSTM model will, thus, allow us to understand how incorporating travel time information would affect the performance of location prediction.

For a baseline model for travel time prediction, we consider a historical mean model that estimates the empirical mean of a travel time for a given location pair, time of day, and day of week based on the training data. More specifically, for each ordered pair of two detector locations k , we combine the travel time observations during hour of day H and day of week W , denoted by $D_k(H, W)$, and average them to obtain the predicted mean travel time, $D_k^*(H, W)$, as follows:

$$D_k^*(H, W) = \frac{1}{n} \sum D_k(H, W) \quad (18)$$

where n is the number of occurrences of $D_k(H, W)$ in the training data.

Besides the historical mean model, we also test a deep learning model based on LSTM with self-attention, referred to as Attention-LSTM-T (Attention-LSTM for travel time prediction only). Fig. 8 shows the model architecture of Attention-LSTM-T, which is similar to

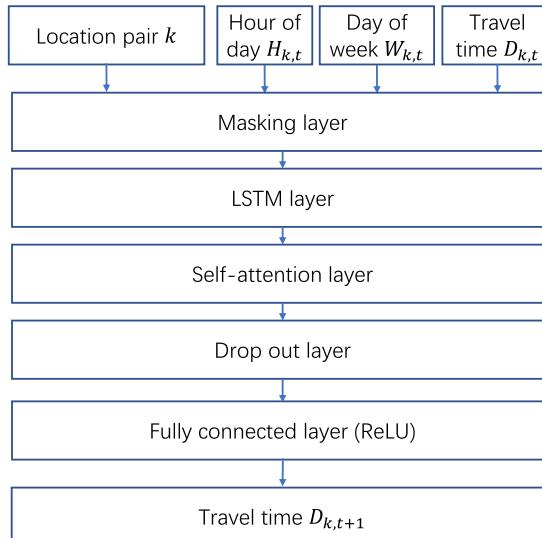


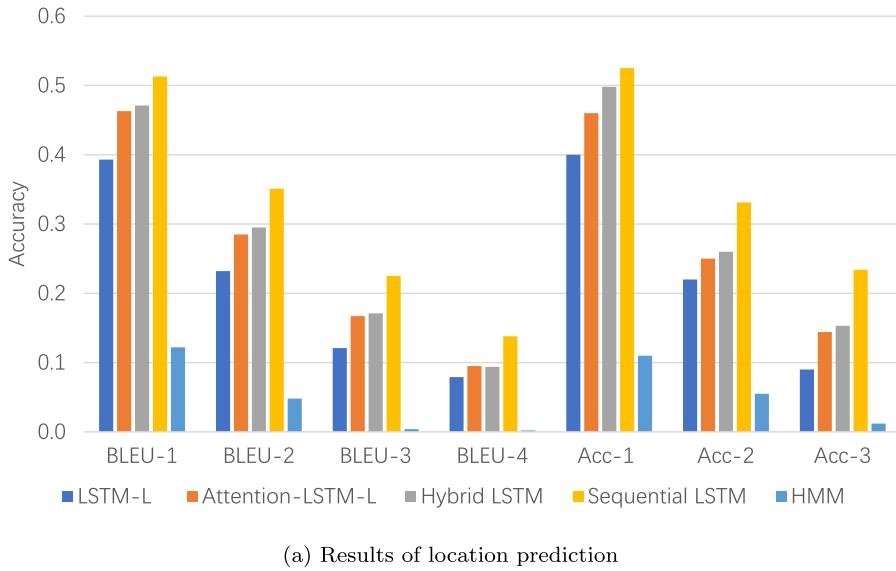
Fig. 8. Structure of Attention-LSTM-T (baseline model for travel time prediction).

the travel time prediction part of our proposed hybrid and sequential LSTM models. A major difference is that Attention-LSTM-T is *detector-based*, whereas the hybrid and sequential LSTM models are *trajectory-based*, meaning that Attention-LSTM-T predicts the time-series of travel time for a given detector pair as in many conventional traffic time-series prediction models, whereas the hybrid and sequential LSTM models predict travel time along the different detector pairs passed by a given trajectory. For each ordered pair of two detector locations k , we construct 15-min aggregated time-series of travel time and use these time-series data from all detector pairs as input sequences to train the Attention-LSTM-T model. At each 15-min time interval t , Attention-LSTM-T predicts the travel time for detector pair k in the next time interval $t + 1(D_{k,t+1})$ based on detector pair k , hour of day $H_{k,t}$, day of week $W_{k,t}$, and current 15-min average travel time $D_{k,t}$ as shown in Fig. 8.

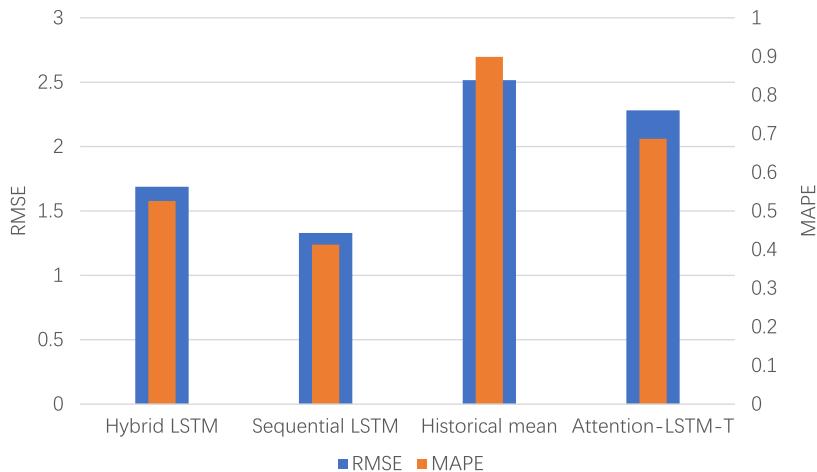
5.3. Evaluation results

The trained models are applied on the testing dataset and the results are shown in Fig. 9, where Fig. 9a shows the location prediction results ($BLEU_N$ is labelled as BLEU- N) and Fig. 9b shows the travel time prediction results (Acc_n is labelled as Acc- n). As we can see from Fig. 9a, the prediction accuracy of LSTM-based models is significantly higher than that of HMM, in terms of both $BLEU_N$ and Acc_n scores (the higher the score, the better the performance), which demonstrates the superior performance of LSTM-based model than HMM in capturing long-range dependencies between different locations. Take BLEU-1 and Acc-1 scores as example; those are all above (or around) 0.4 for LSTM-based models, while those for HMM are just above 0.1.

In addition, with the incorporation of self-attention layer, Attention-LSTM-L outperforms LSTM-L, showing the effective of self-



(a) Results of location prediction



(b) Results of travel time prediction

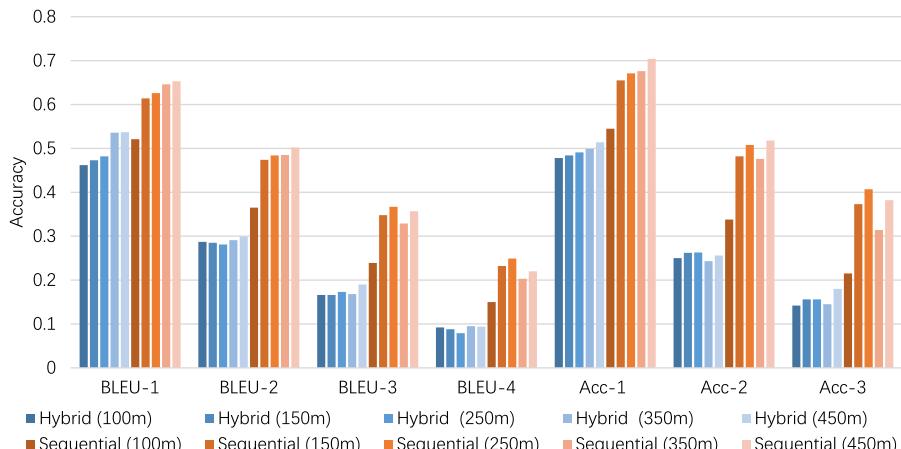
Fig. 9. Evaluation results of different models.

attention layer in sequential learning. By comparing the results between Attention-LSTM-L and the hybrid and sequential LSTM models for joint location-time prediction, it is obvious that the travel time prediction helps the prediction of next location. One possible explanation could be that en-route travel times reflect the current network traffic conditions, which tend to affect a driver's route choice decisions (e.g., a driver may decide to detour if he/she has experienced longer travel time along the way), and incorporating travel time sequence information, thus, helps the prediction of location sequence. Between the two proposed models, the sequential LSTM performs better than the hybrid LSTM as indicated by higher BLEU and Acc scores (e.g., the BLEU-1 score is 0.521 for the sequential LSTM and 0.462 for the hybrid LSTM). This suggests that learning separate parameter sets for location and time prediction tasks while still passing the information between the two tasks is a better approach than sharing parameters.

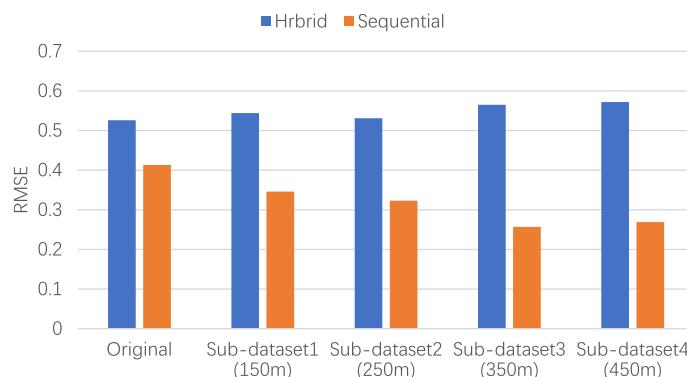
The travel time prediction results are shown in Fig. 9b. The proposed hybrid and sequential LSTM models achieve higher accuracy showing the RMSE of 1.7 min and 1.3 min (MAPE of 52.6% and 41.3%), respectively, than the baseline historical mean and Attention-LSTM-T models, which have the RMSE of 2.5 min and 2.3 min (MAPE of 89.9% and 68.7%), respectively (the lower the value, the better the performance). The historical mean model shows the highest errors and the comparison between the historical mean and Attention-LSTM-T models suggests the importance of incorporating the current travel time information in predicting travel times in a dynamic traffic network as in Attention-LSTM-T, as opposed to relying on the static historical average as in the historical mean model. Between the two proposed models, the sequential LSTM outperforms the hybrid LSTM, which further demonstrates the superiority of the sequential LSTM structure that can capture dependency between route choice decision and travel time experience while allowing separate parameter sets to be optimised for different tasks of location and time prediction.

5.4. The impact of spatial resolution

We further investigate the impact of spatial resolution of trajectory data on the model performance. The motivation of this analysis is to understand how densely detection points should be distributed in the network to ensure the trajectory prediction with a desired level of accuracy, which can offer insights into the optimal detector density for mobility monitoring and prediction. Four detector



(a) Results of location prediction



(b) Results of travel time prediction

Fig. 10. Evaluation results with different spatial resolutions.

subsets with different minimum detector distances (i.e., 150 m, 250 m, 350 m, and 450 m) are studied and compared with the original detector network. The prediction results of different spatial resolutions with the two proposed LSTM models are presented in Fig. 10 and Table 3. In Fig. 10a, model type (Hybrid or Sequential) and minimum detector distance are shown as the name for each test case (100 m is used to indicate the original detector network).

As shown in Fig. 10a, with the increase of minimum distance between detectors and decrease in spatial resolution, the accuracy of next location prediction tends to be improved for both the sequential LSTM and hybrid LSTM. This is particularly true for short-range prediction (i.e., predicting the immediate next location) measured by BLEU-1 and Acc-1. One possible explanation can be that, when the distance between detectors is larger, it would be easier for a model to narrow down which detectors a vehicle can go next from the current point and, thus, it leads to higher prediction accuracy. This spatial resolution effect may be less helpful for long-range prediction (i.e., predicting more than one next locations) as the increasing trend is less apparent for BLEU-2,3,4 and Acc-2,3. In fact, there is a trend that predicting more than two consecutive locations performs best under the network with the minimum detector distance of 250 m, as indicated by the BLEU-3, BLEU-4, and Acc-3 values peaked for the 250-m case. Between the two models, the sequential LSTM always performs better than the hybrid LSTM (e.g., the best BLEU-1 scores for the hybrid LSTM and sequential LSTM are with minimum distance of 450 m, which are 0.537 and 0.653, respectively).

For comparing the travel time prediction results under different spatial resolutions, only MAPE is used as performance measure because the travel time between two locations increases as the distance increases and the RMSEs for lower spatial resolution data (larger detector distances) are inherently higher than those for higher spatial resolution data, thereby making RMSEs not comparable across different sub-datasets. As shown in Fig. 10b, the MAPE for the hybrid LSTM is generally increasing (the prediction accuracy is decreasing) with the increase in minimum detector distance, whereas the MAPE for the sequential LSTM is decreasing (the prediction accuracy is increasing). The results indicate that the sequential LSTM model can achieve good results for both location prediction and travel time prediction—performance increase in location prediction may actually help time prediction—, while the hybrid LSTM model compromises between the two prediction tasks—increase in location prediction performance may come at the expense of time prediction accuracy as the shared parameters are learned more toward location prediction. It is noted that there might be other factors that affect the accuracy of travel time prediction under different spatial resolutions such as systematic changes in data variation. For instance, we measured the variation of travel time data in terms of *the coefficient of variation* (the ratio of the standard deviation to the mean) and found that there is a tendency, although weak, that the variation in travel time measurement appears to decrease as the travel distance used for measurement increases. Such a decrease in data variation may help increase prediction accuracy and, thus, could partially account for the increasing prediction accuracy with increasing detector distance in the sequential LSTM model. The decreasing accuracy in the hybrid LSTM model would then indicate a poorer performance of the hybrid model even after taking into account the effect of data variation. Our analysis results, however, were not strong enough to support this effect and further research is needed to confirm the systematic change in travel time data variation with respect to spatial resolution and its impact on prediction accuracy.

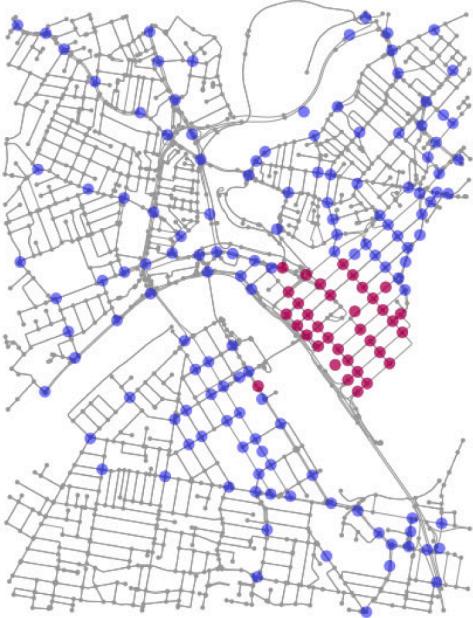
One concern about the previous analysis might be whether the performance increase with respect to detector distance increase is because there are fewer detection points in the study network and, thus, it is easier to predict the next locations. To remove the influence of detector set size and trip length, we select 35 detectors from each network and perform the test again with trajectory data from these sub-networks with the equal detector set size. The sub-networks with the minimum detector distance of 100 m, 150 m, 250 m, and 350 m are shown in Fig. 11 (red points are the 35 selected detectors). After controlling the detector set size, we still obtain the similar results to the previous analysis as shown in Fig. 12 and Table 4, i.e., the sequential LSTM outperforms the hybrid LSTM; for short-range prediction the accuracy increases with the increase of detector distance, while for long-range prediction the accuracy first increases and then decreases.

6. Application in network traffic flow analysis

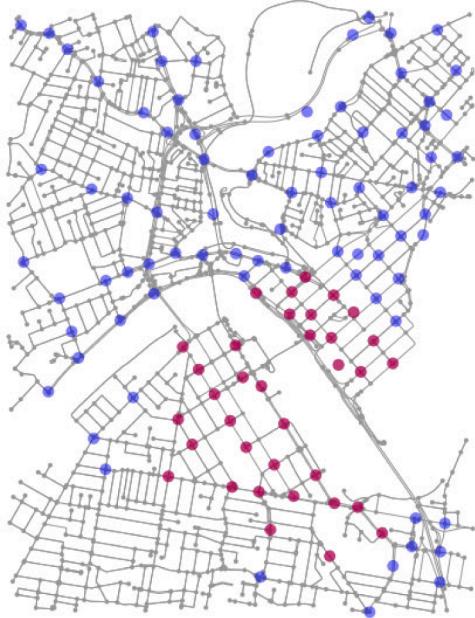
In this section, we demonstrate an application of the proposed models in analysing and managing network traffic. We use the proposed models to predict the traffic flow connectivity between two locations. For a given target location of interest, c , such as congestion hot spot, we are interested in identifying location i that has high flow connectivity with c (e.g., locations that send most traffic flows to c) to better design traffic control and management strategies to mitigate congestion. We define two connectivity

Table 3
Evaluation results with different spatial resolutions.

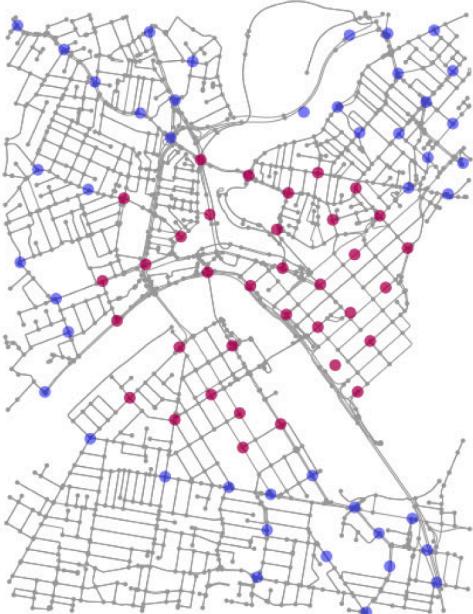
	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Acc-1	Acc-2	Acc-3	MAPE
Hybrid (100 m)	0.462	0.287	0.166	0.092	0.478	0.25	0.142	0.526
Hybrid (150 m)	0.473	0.285	0.166	0.088	0.484	0.262	0.156	0.544
Hybrid (250 m)	0.482	0.281	0.173	0.079	0.491	0.263	0.156	0.531
Hybrid (350 m)	0.536	0.291	0.168	0.095	0.499	0.243	0.145	0.565
Hybrid (450 m)	0.537	0.299	0.190	0.094	0.514	0.256	0.18	0.572
Sequential (100 m)	0.521	0.365	0.239	0.150	0.545	0.338	0.215	0.413
Sequential (150 m)	0.614	0.474	0.348	0.232	0.655	0.482	0.373	0.346
Sequential (250 m)	0.626	0.484	0.367	0.249	0.671	0.508	0.407	0.323
Sequential (350 m)	0.646	0.485	0.329	0.203	0.676	0.476	0.314	0.257
Sequential (450 m)	0.653	0.502	0.357	0.220	0.704	0.518	0.382	0.269



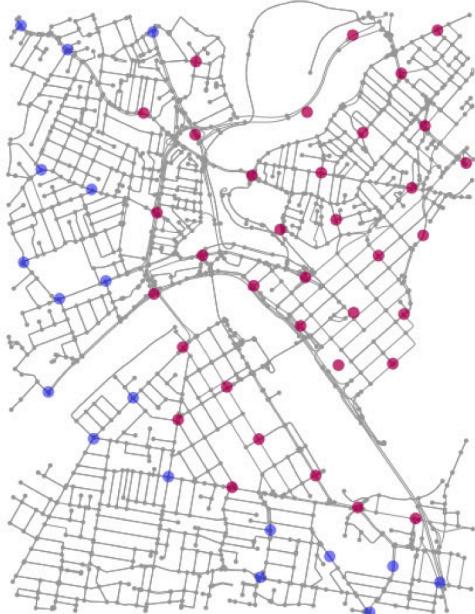
(a) Minimum Distance=100m



(b) Minimum Distance=150m



(c) Minimum Distance=250m



(d) Minimum Distance=350m

Fig. 11. Sub-networks with the fixed number of detectors (red points represent the 35 selected detectors for each network).

measures: the outflow rate at location i with respect to critical location c , denoted by $R_{i,out}$, which is the proportion of all the vehicles passing i that arrive in c , and the inflow rate at critical location c with respect to other location i , denoted by $R_{i,in}$, which is the proportion of all the vehicles arriving in c that came from i . These are shown in Eqs. (19) and (20):

$$R_{i,out} = \frac{n_{i,c}}{n_i} \quad (19)$$

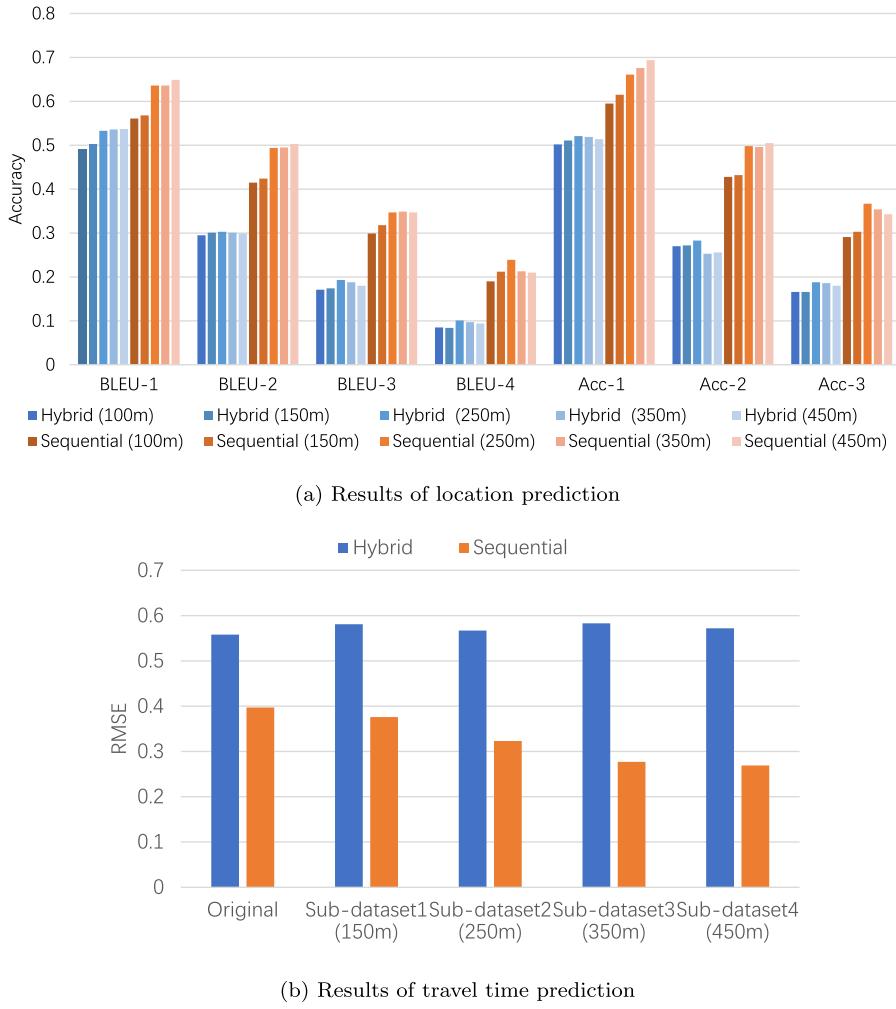


Fig. 12. Evaluation results with different spatial resolutions (with the fixed number of detectors).

Table 4

Evaluation results with different spatial resolutions (with the fixed number of detectors).

	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Acc-1	Acc-2	Acc-3	MAPE
Hybrid (100 m)	0.491	0.295	0.171	0.085	0.502	0.27	0.166	0.558
Hybrid (150 m)	0.503	0.301	0.174	0.084	0.511	0.272	0.166	0.581
Hybrid (250 m)	0.533	0.303	0.193	0.101	0.521	0.283	0.188	0.567
Hybrid (350 m)	0.536	0.301	0.188	0.097	0.519	0.253	0.186	0.583
Hybrid (450 m)	0.537	0.299	0.180	0.094	0.514	0.256	0.18	0.572
Sequential (100 m)	0.561	0.415	0.299	0.190	0.595	0.428	0.291	0.397
Sequential (150 m)	0.568	0.424	0.318	0.212	0.615	0.432	0.303	0.376
Sequential (250 m)	0.636	0.494	0.347	0.239	0.661	0.498	0.367	0.323
Sequential (350 m)	0.636	0.495	0.349	0.213	0.676	0.496	0.354	0.277
Sequential (450 m)	0.649	0.503	0.347	0.209	0.694	0.505	0.343	0.269

$$R_{i,c} = \frac{n_{i,c}}{n_c} \quad (20)$$

where n_i and n_c are the number of vehicles that traverse source location i and target location c , respectively, and $n_{i,c}$ denotes the number of vehicles that travel from i to c (not necessarily directly, i.e., vehicles visiting other locations between i and c are also included).

For the sake of demonstration, we use the network with minimum distance of 250 m as an example here. We first identify the critical location that has the highest traffic volume based on the Bluetooth vehicle trajectories on 1 June 2016, which is shown as the blue point in Fig. 13. We then generate vehicle trajectories with the proposed models as well as two baseline models (i.e., LSTM-L and

HMM) to predict the outflow and inflow rates between different locations and the critical location. The actual and predicted outflow rates of all other locations with respect to the critical location are shown as red points in Fig. 13 (circle size represents relative magnitude), while the actual and predicted inflow rates of critical location with respect to all other locations are shown as purple points in Fig. 14. Note that the circle size is normalised within each model. We can see that the predicted flow connectivity patterns for each location with the LSTM models closely match the observed ones, whereas the HMM shows high discrepancy between the predicted and actual patterns.

The difference between the observed and predicted outflow/inflow rates are further evaluated using the RMSE defined as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (R_i^* - R_i)^2} \quad (21)$$

where n is the number of locations (excluding the critical location) and R_i^* and R_i are the predicted and observed outflow/inflow rates for location i , respectively. The RMSE results are presented in Table 5, where we show the RMSEs not only for all locations besides critical location (RMSE-all), but also for the top 10 locations with the highest probability travelling to critical location (RMSE-10) as they have great influence on the traffic state of the critical location. The sequential LSTM produces the highest accuracy for all the cases, followed by the hybrid LSTM, LSTM-L, and HMM.

The prediction results can be used to draw useful insights into the importance of locations in traffic management. The locations with high inflow rates are the locations that are highly affected by the traffic situation in the critical location because most of the

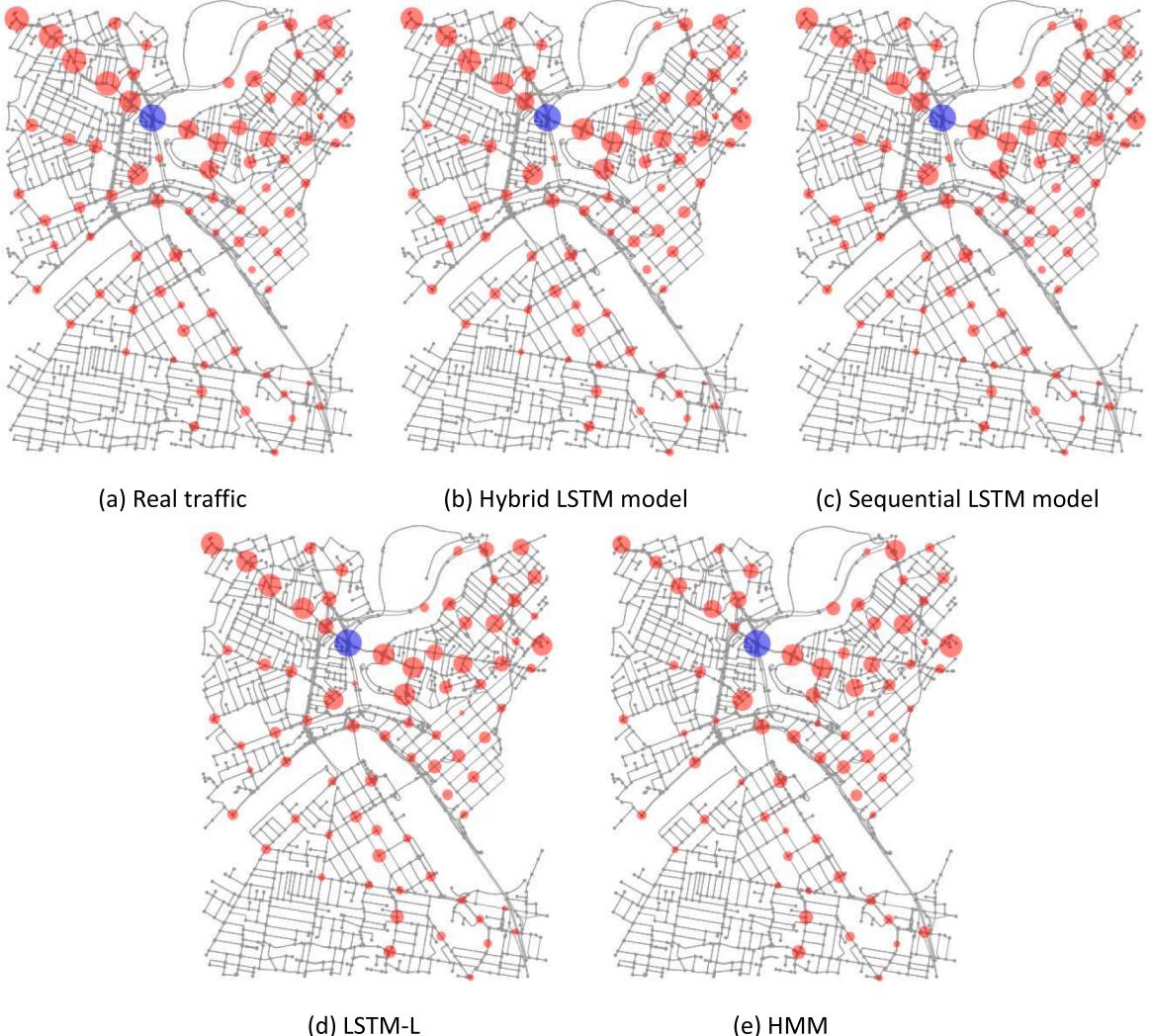


Fig. 13. Outflow rates at other locations (red) with respect to the critical location (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

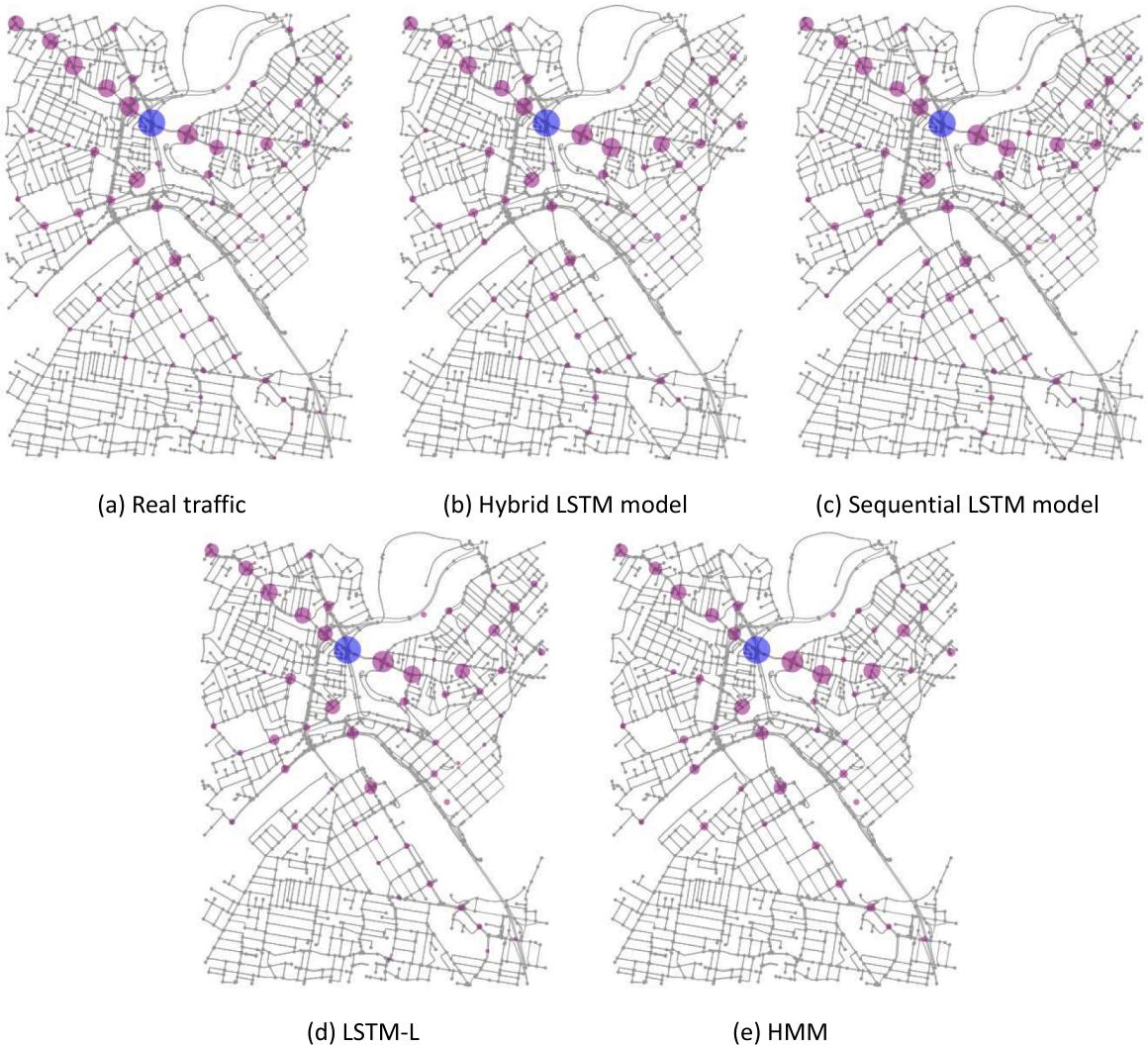


Fig. 14. Inflow rates at the critical location (blue) with respect to other locations (purple). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5

Prediction results of flow connectivity between critical and other locations based on the proposed trajectory prediction models.

Model	Outflow rate		Inflow rate	
	RMSE-all	RMSE-10	RMSE-all	RMSE-10
Hybrid LSTM	0.0316	0.0616	0.0203	0.0491
Sequential LSTM	0.0201	0.0409	0.0170	0.0405
LSTM-L	0.0386	0.0766	0.0223	0.0518
HMM	0.0795	0.1269	0.0433	0.0856

vehicles in those locations need to go to the critical location. When there is a disruption in the critical location, traffic managers may want to focus on these locations to provide detour guidance. On the other hand, the locations with high outflow rates are the locations that send the most traffic to the critical location and, thus, have a high degree of influence on the critical location. To prevent congestion at the critical location, traffic managers may focus on controlling outflows from these locations. The patterns in Figs. 13 and 14 show that locations with high inflow rates and locations with outflow rates do not necessarily overlap. Therefore, this kind of analysis on trajectory-based flow connectivity can support the identification of important locations for different traffic management purposes, which complement the knowledge of static characteristics such as distance, physical connectivity, and network structure.

7. Conclusions

This paper develops models to predict the trajectory of an individual vehicle—both next locations and arrival times—using state-of-the-art deep learning methods. Two structures of deep learning models were proposed, namely hybrid LSTM model and sequential LSTM model, to capture the spatial-temporal dependencies between different locations in a network based on LSTM and self-attention.

The Bluetooth detector data from the downtown area of Brisbane, Australia were used to evaluate the performance of the proposed models. The evaluation results show that the prediction accuracy of LSTM-based deep learning models are significantly higher than that of traditional models such as HMM for location prediction and historical mean for time prediction. The incorporation of self-attention mechanism further improves prediction accuracy. The incorporation of travel time prediction in the location prediction models (i.e., jointly modeling location and time prediction) improves the location prediction performance, compared to the location-only prediction models without travel time prediction components. Between the two proposed models, the sequential LSTM model outperforms the hybrid LSTM model in both location and travel time prediction, which demonstrates the advantage of learning separate parameter sets for location and time prediction tasks while still passing the information between the two tasks through sequentially linked two LSTM networks, compared to sharing input and parameters in a single hybrid LSTM network.

Overall, the framework of jointly predicting location and time together—and thus incorporating travel time information as input—appears to be beneficial in capturing the information on the surrounding traffic and consequently enhancing the prediction accuracy for both location and travel time prediction tasks. Although our framework treats each trajectory as a separate input and does not explicitly model a group of trajectories or their interactions, our models still capture the influence of the surrounding traffic through the incorporation of previous travel time observations as input sequence. More specifically, in our framework, the future location and travel time are assumed to not only depend on just the previous locations and temporal information (time-of-day and day-of-week) but also depend on the vehicle's experienced travel times along the previously visited locations. For instance, if a vehicle has experienced a longer travel time along its journey so far, this may affect the vehicle's route choice to seek an alternative route. It may also affect the future travel times as it is more likely that the travel time ahead would be also longer because the network is currently congested. Since these experienced travel times reflect the consequences of vehicle interactions and the prevailing congestion levels on the roads, using that as input allows us to incorporate the dynamic condition of network traffic into travel time prediction, even if we do not explicitly include aggregate traffic flow parameters (such as link volume, density, or speed) as input.

The impact of spatial resolution of trajectory data (the density of detectors in a network) on the model performance was also investigated. The results show that the spatial resolution of trajectory data has noticeable impact on the model performance, where the prediction accuracy increases with the increase in the distance between locations in trajectories, particularly for short-range prediction (i.e., predicting the immediate next location). The impact is less clear for long-range prediction (i.e., predicting more than one next locations).

To demonstrate the application of the proposed models in the context of traffic management, we designed an experiment to identify important locations for traffic control to mitigate congestion in a hot-spot by utilising the capability to predict the next locations of vehicles. We measured the connectivity between locations in terms of the number of vehicles travelling between them based on the predicted trajectories generated by the proposed models. The results show that the two proposed models, especially the sequential LSTM model, can closely reproduce the actual flow connectivity patterns, showing where vehicles come from and what locations are important in controlling and mitigating traffic congestion at a given hot-spot in the network.

Acknowledgements

This research was partially funded by the Australian Research Council under grant DE190101020. The authors would like to thank Department of Transport and Main Roads and Brisbane City Council for access to the data on which the paper is based. The authors of course remain solely responsible for the content of this paper.

References

- Abedi, N., Bhaskar, A., Chung, E., Miska, M., 2015. Assessment of antenna characteristic effects on pedestrian and cyclists travel-time estimation based on Bluetooth and WiFi MAC addresses. *Transportation Research Part C: Emerging Technologies* 60, 124–141.
- Alhasoun, F., Alhazzani, M., Aleissa, F., Alnasser, R., González, M., 2017. City scale next place prediction from sparse data through similar strangers. In: Proceedings of ACM KDD Workshop, pp. 191–196.
- Amirrudin, N.A., Ariffin, S.H., Malik, N.A., Ghazali, N.E., 2013. User's mobility history-based mobility prediction in LTE femtocells network. In: 2013 IEEE International RF and Microwave Conference (RFM). IEEE, pp. 105–110.
- Ashbrook, D., Starner, T., 2002. Learning significant locations and predicting user movement with GPS. In: Proceedings. Sixth International Symposium on Wearable Computers. IEEE, pp. 101–108.
- Ashbrook, D., Starner, T., 2003. Using GPS to learn significant locations and predict movement across multiple users. *Pers. Ubiquit. Comput.* 7 (5), 275–286.
- Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- Baratchi, M., Meratnia, N., Havinga, P.J., Skidmore, A.K., Toxopeus, B.A., 2014. A hierarchical hidden semi-Markov model for modeling mobility data. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 401–412.
- Bhaskar, A., Chung, E., 2013. Fundamental understanding on the use of bluetooth scanner as a complementary transport data. *Transportation Research Part C: Emerging Technologies* 37, 42–72.
- Carpenter, C., Fowler, M., Adler, T.J., 2012. Generating route-specific origin-destination tables using Bluetooth technology. *Transp. Res. Rec.* 2308 (1), 96–102.
- Chen, M., Yu, X., Liu, Y., 2019. MPE: A mobility pattern embedding model for predicting next locations. *World Wide Web* 22 (6), 2901–2920.
- Chen, M., Zuo, Y., Jia, X., Liu, Y., Yu, X., Zheng, K., 2020a. CEM: A convolutional embedding model for predicting next locations. *IEEE Transactions on Intelligent Transportation Systems* (in press) 1–10.

- Chen, Y., Long, C., Cong, G., Li, C., 2020b. Context-aware deep model for joint mobility and time prediction. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 106–114.
- Cheng, J., Dong, L., Lapata, M., 2016. Long short-term memory-networks for machine reading. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 551–561.
- Choi, S., Kim, J., Yeo, H., 2019. Attention-based recurrent neural network for urban vehicle trajectory prediction. *Procedia Computer Science* 151, 327–334.
- Choi, S., Yeo, H., Kim, J., 2018. Network-wide vehicle trajectory prediction in urban traffic networks using deep learning. *Transp. Res. Rec.* 2672 (45), 173–184.
- Do, T.M.T., Gatica-Perez, D., 2012. Contextual conditional models for smartphone-based human mobility prediction. In: Proceedings of the 2012 ACM conference on ubiquitous computing, pp. 163–172.
- Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L., 2016. Recurrent marked temporal point processes: Embedding event history to vector. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1555–1564.
- Duives, D.C., Wang, G., Kim, J., 2019. Forecasting pedestrian movements using recurrent neural networks: An application of crowd monitoring data. *Sensors* 19 (2).
- Endo, Y., Nishida, K., Toda, H., Sawada, H., 2017. Predicting destinations from partial trajectories using recurrent neural network. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 160–172.
- Gambs, S., Killijian, M.-O., del Prado Cortez, M.N., 2012. Next place prediction using mobility Markov chains. In: Proceedings of the First Workshop on Measurement, Privacy, and Mobility, pp. 1–6.
- Gers, F.A., Schraudolph, N.N., Schmidhuber, J., 2002. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research* 3 (Aug), 115–143.
- Gidófalvi, G., Dong, F., 2012. When and where next: individual mobility prediction. In: Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, pp. 57–64.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Huang, X., Sun, J., Sun, J., 2018. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transportation Research Part C: Emerging Technologies* 95, 346–362.
- Ivanovic, B., Leung, K., Schmerling, E., Pavone, M., 2020. Multimodal deep generative models for trajectory prediction: A conditional variational autoencoder approach. *IEEE Robotics and Automation Letters* 6 (2), 295–302.
- Jain, A., Casas, S., Liao, R., Xiong, Y., Feng, S., Segal, S., Urtasun, R., 2020. Discrete residual flow for probabilistic pedestrian behavior prediction. In: Conference on Robot Learning. PMLR, pp. 407–419.
- Kim, J., Mahmassani, H.S., 2015. Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories. *Transportation Research Part C: Emerging Technologies* 59, 375–390.
- Krishna, K., Jain, D., Mehta, S.V., Choudhary, S., 2018. An LSTM based system for prediction of human activities with durations. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1 (4), 1–31.
- Kulkarni, V., Garbinato, B., 2019. 20 years of mobility modeling & prediction: Trends, shortcomings & perspectives. In: Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 492–495.
- Laharotte, P.-A., Billot, R., Come, E., Oukhellou, L., Nantes, A., El Faouzi, N.-E., 2015. Spatiotemporal analysis of Bluetooth data: Application to a large urban network. *IEEE Trans. Intell. Transp. Syst.* 16 (3), 1439–1448.
- Li, M., Lu, F., Zhang, H., Chen, J., 2020. Predicting future locations of moving objects with deep fuzzy-LSTM networks. *Transportmetrica A: Transport Science* 16 (1), 119–136.
- Li, S., Li, W., Cook, C., Zhu, C., Gao, Y., 2018. Independently recurrent neural network (IndRNN): Building a longer and deeper RNN. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5457–5466.
- Liu, Q., Wu, S., Wang, L., Tan, T., 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In: Thirtieth AAAI conference on artificial intelligence.
- Liu, Q., Zuo, Y., Yu, X., Chen, M., 2019. TTDM: A travel time difference model for next location prediction. In: 2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, pp. 216–225.
- Lu, X., Wetter, E., Bharti, N., Tatem, A.J., Bengtsson, L., 2013. Approaching the limit of predictability in human mobility. *Scientific Reports* 3, 2923.
- MacDonald, I.L., Zucchini, W., 1997. Hidden Markov and other models for discrete-valued time series. CRC Press.
- Mathew, W., Raposo, R., Martins, B., 2012. Predicting future locations with hidden Markov models. In: Proceedings of the 2012 ACM conference on ubiquitous computing, pp. 911–918.
- Michau, G., Nantes, A., Bhaskar, A., Chung, E., Abry, P., Borgnat, P., 2017. Bluetooth data in an urban context: Retrieving vehicle trajectories. *IEEE Trans. Intell. Transp. Syst.* 18 (9), 2377–2386.
- Monreale, A., Pinelli, F., Trasarti, R., Giannotti, F., 2009. Wherenext: a location predictor on trajectory pattern mining. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 637–646.
- Noulas, A., Scellato, S., Lathia, N., Mascolo, C., 2012. Mining user mobility features for next place prediction in location-based services. In: 2012 IEEE 12th international conference on data mining. IEEE, pp. 1038–1043.
- Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318.
- Pathirana, P.N., Savkin, A.V., Jha, S., 2003. Mobility modelling and trajectory prediction for cellular networks with mobile base stations. In: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pp. 213–221.
- Rathore, P., Kumar, D., Rajasegarar, S., Palaniwami, M., Bezdek, J.C., 2019. A scalable framework for trajectory prediction. *IEEE Trans. Intell. Transp. Syst.* 20 (10), 3860–3874.
- Scellato, S., Musolesi, M., Mascolo, C., Latora, V., Campbell, A.T., 2011. Nextplace: a spatio-temporal prediction framework for pervasive systems. In: International Conference on Pervasive Computing. Springer, pp. 152–169.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15 (1), 1929–1958.
- Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning 4 (2), 26–31.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008.
- Wang, T.-H., Manivasagam, S., Liang, M., Yang, B., Zeng, W., Urtasun, R., 2020. V2VNet: Vehicle-to-vehicle communication for joint perception and prediction. In: European Conference on Computer Vision. Springer, pp. 605–621.
- Yavaş, G., Katsaros, D., Ulusoy, Ö., Manolopoulos, Y., 2005. A data mining approach for location prediction in mobile environments. *Data & Knowledge Engineering* 54 (2), 121–146.
- Ying, J.J.-C., Lee, W.-C., Tseng, V.S., 2014. Mining geographic-temporal-semantic patterns in trajectories for location prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5 (1), 1–33.
- Zhang, R., Guo, J., Jiang, H., Xie, P., Wang, C., 2019. Multi-task learning for location prediction with deep multi-model ensembles. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, pp. 1093–1100.
- Zheng, G., Mukherjee, S., Dong, X.L., Li, F., 2018. Opentag: Open attribute value extraction from product profiles. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1049–1058.