Multi-Semantic Path Representation Learning for Travel Time Estimation

Liangzhe Han, Bowen Du, *Member, IEEE*, Jingjing Lin, Leilei Sun, *Member, IEEE*, Xucheng Li, and Yizhou Peng

***Abstract*— Travel time estimation of a given path is a crucial task of Intelligent Transportation Systems (ITS). Accurate travel time estimation can benefit multiple downstream applications such as route planning, real-time navigation, and urban construction. However, it is a challenging problem since the travel time is largely affected by multiple complicated factors including spatial factors, temporal factors and external factors, and obtaining informative representations of a given path is not trivial. Most previous works solved this problem in either Euclidean space or non-Euclidean space, which was unilateral to represent the actual traveling path and led to relatively poor performance. To address this, this paper proposes a multi-semantic path representation method to exploit information in Euclidean space and non-Euclidean space simultaneously. First, since the path is composed of several segments, we generate semantic representations of segments in non-Euclidean space by taking both the time information and the historical co-occurrence into consideration. Second, as the path could be equally represented as several travelled intersections, semantic representations of intersection sequences are also extracted to improve the capability of the method by considering information in Euclidean space. Meanwhile, semantic representations from properties, including the length and the type of segments, are also incorporated into the model. Finally, a sequence learning component is added on the top to aggregate the information along the entire path and provides the final estimation. Extensive experiments were conducted on two real-world taxi trajectories datasets, and the experimental results demonstrate the superiority of the proposed method.**

***Index Terms*— Travel time estimation, sequence learning, semantic representation.**

Manuscript received 25 February 2021; revised 24 July 2021 and 30 August 2021; accepted 15 September 2021. Date of publication 20 October 2021; date of current version 9 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 51822802, Grant 51991395, Grant 71901011, and Grant U1811463; in part by the Science and Technology Major Project of Beijing under Grant Z191100002519012; in part by the National Key Research and Development Program of China under Grant 2018YFB2101003; in part by the China Geological Survey under Grant DD20190637; and in part by the Guangxi Innovation-Driven Development Special Fund Project under Grant AA18118053. The Associate Editor for this article was Y. Lv. *(Corresponding author: Leilei Sun.)*

Liangzhe Han, Bowen Du, and Leilei Sun are with the State Key Laboratory of Software Development Environment (SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: liangzhehan@buaa.edu.cn; dubowen@buaa.edu.cn; leileisun@buaa.edu.cn).

Jingjing Lin is with the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing 100191, China (e-mail: linjingjing@buaa.edu.cn).

Xucheng Li and Yizhou Peng are with Shenzhen Urban Transport Planning Center Company Ltd., Shenzhen 518057, China (e-mail: xucheng.li@sutpc.com; pengyizhou@sutpc.com).

Digital Object Identifier 10.1109/TITS.2021.3119887

I. Introduction

With the development of metropolitan and the increasing commuting demand of citizens, Intelligent Transportation System plays a more and more important role in people's daily life. Among Intelligent Transportation System, travel time estimation is a fundamental task due to the extensive use of vehicles and the tight relation with several most common applications including route planning, real-time navigation and urban construction.

An accurate estimation of the travel time can give practical advice about how to save time and thus benefits the whole society. Specifically, it takes an origin-destination (OD) pair with or without the middle path at a leaving time as input and estimates the time spent in travelling through the path.

Previous works on travel time estimation can be roughly divided into two categories: OD based methods and path based methods. OD based methods [1]–[3] mainly considered geography coordinates of origin and destination. These methods generally estimated the travel time of the current query based on historical trajectories with adjacent origin and destination while they omitted the information in the middle of a path. Path based methods took the information of the whole path into consideration, and it led to the categorization about how to represent a path. As shown in Fig. 1, a same path could be represented in multiple forms which fall into Euclidean space or non-Euclidean space. Some works [4] represented a path as a sequence of road segments and summed up predicted travel time of each single segment. The assumption that the final travel time is the sum of segments' travel time overlooked the transaction time between segments. Though improvement has been made to choose optimal sub-path instead of single segment, the performance of these methods is still limited. With the development of deep learning, there were some works taking the whole path as input to estimate the travel time. Since trajectories captured by GPS devices are generally saved as sequences of GPS points, it is natural to handle a path as a sequence of GPS points [5]. Some work [6] took a further step to map GPS points to grids and exploited historical statistic information about grids. It is also remarkable to represent a path as a sequence of segments [7], [8]. In summary, OD-based, grids-based and GPS-based methods solved the problem in Euclidean space, and segments-based methods solved the problem in non-Euclidean space.

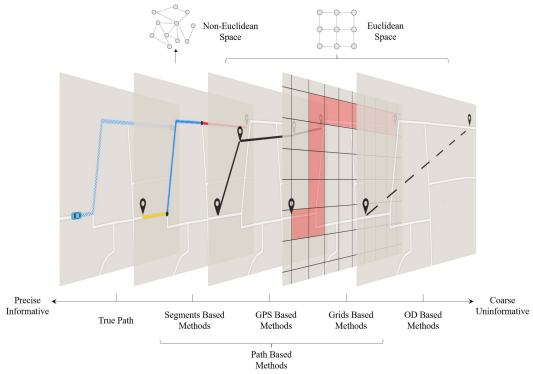


Fig. 1. Different methods to represent a path.

Though plenty of works extracted the information from recorded data as much as possible, they either represented the path in a coarse way or processed the path in unilateral space, which limited their performance. Thus, it is still a challenging task due to the following reasons: First, a path always has multiple types of spatial features. It is apparent that a long path would take a long time indicating that features in Euclidean space matters. Meanwhile, features in non-Euclidean space which indicate positions in the road network are also responsible for the travel time. Second, the travel time varies across time due to people's activities. For instance, travelling through the same segment takes more time in rush hour than that in free hour, and the difference also exists between weekdays and weekends. Third, the travel time can be affected by multiple properties features including whether a segment is a highway.

To this end, we propose a multi-Semantic model for Travel Time Estimation (STTE), which leverages multiple types of features and extracts multiple semantic representations of a path for accurate travel time estimation. First, road segments are organized as a network in real world so how to represent positions of segments in road network is important to travel time estimation. Semantic representations of segments contains information about how people tend to travel from one to another, which exists in historical trajectories records and peoples' schedule. Thus considering co-occurrence of segments in historical trajectories and the time semantic information of the query, we design a component for spatial semantic representations in non-Euclidean space. Second, spatial features in Euclidean space could reflect the distance of spatial movement and are crucial for travel time estimation. Semantic representations of these features could be obtained from travelled intersections; this also provides another view of a path. Then, to represent a path in a informative way, we fuse semantic representations of properties, time-aware segments' semantic representations in non-Euclidean space and intersections' semantic representations in Euclidean space together. And finally, a sequence learning component is added to aggregate information along the entire path and estimate the travel time. In summary, our contributions are three folds:

- To represent a path, we leverage multiple features of a given path. The input path is not only as a sequence of segments to provide semantic representations in non-Euclidean space but also as a sequence of intersections to provide semantic representations in Euclidean space.
- The time information and the relationship of co-occurrence of segments are combined together to get time-aware semantic representations of segments. It ensures segments which often occur together have tighter relationship and the segments would perform differently at different time (e.g., weekdays and weekends).
- The proposed model is evaluated by extensive experiments on two large real-world datasets. And the comparison is made with multiple existing travel time estimation methods including machine learning methods, OD-based methods, GPS-based methods and segment-based methods. The experimental results demonstrate the superiority of our proposed method.

The structure of this paper is as follows: Section II reviews related work; Section III formalizes the task and introduces what data is involved; Section IV provides details about every component of our model; Section V introduces settings and results of experiments; Section VI concludes our work.

II. Related Work

A. Travel Time Estimation

In the field of travel time estimation, previous work can be roughly divided into two categories: origin-destination based methods and path based methods.

If the input data only consists of origin and destination information, origin-destination based methods could be applied. Wang *et al.* [3] estimated the travel time by statistic of historical adjacent trajectories. This method considered travel time estimation in an explicit and heuristic way. Jindal *et al.* [1] proposed a deep neural network to estimate the distance and the travel time simultaneously from coordinates of origin, destination and the leaving time. These origin-destination based methods overlooked the underlying road network structure as well as the spatial-temporal property. To overcome this issue, Li *et al.* [2] leveraged the topological information as well as the spatial-temporal prior knowledge. In general, this kind of approach estimates travel time with historical trips which have a similar spatial and temporal information. However, these methods overlooked extensive information between the origin and the destination which limited the performance of them.

The path-based methods solved the problem by considering every part of a path instead of only origin and destination. The original idea was to calculate travel time on each road segment separately and sum them up as the travel time of the whole path [4]. To get the historical travel time of road segments, two types of data have been used: loop detector data [9]–[12] and floating car data [13], [14]. Multiple methods were used to predict the travel time of a query path. Fabritiis *et al.* [15] estimated segments' travel time by long-term and short-term distribution assumption. Wang *et al.* [16] predicted the travel time by decomposing a tensor with road segments, drivers and the time information. Zygouras *et al.* [17] predicted the future travel time by a novel covariance function and Gaussian processes. It seems natural to sum up the travel time of

each single segment [4], [17]. Nevertheless, segment-based methods fail to take the transition times between segments like waiting for traffic lights and making turns into consideration. To overcome this, sub-path based methods which treated a part of the path containing several road segments as single units were proposed. An important direction of these method is how to determine the optimal sub-path [16]. Sub-path based methods still overlooked a portion of transactions. With the development of deep learning, there were some research that directly utilized the whole path to estimate travel time. And it leads to a new question about how to represent a path.

B. Path Representation

In term of travel path representation, previous work can be divided into three categories: GPS based methods [18], grid sequence based methods and segment based methods. Wang *et al.* [5] used a sequence of consecutive GPS points, containing latitude, longitude and timestamp to represent a path. They combined GPS points and some external factors, such as the time, the weather condition, day of week and the corresponding driver into a vector for training. The GPS based methods could leverage information in the middle of path, but they failed to capture positions of a path in road network. Meanwhile, their performance highly relied on the frequency and the quality of raw GPS data. Zhang *et al.* [6] partitioned the concerned area into several disjoint but equal-sized grids, and they assumed that as long as the granularity of grid cells is fine enough, the grids can capture the real movement of the path in road networks. Though multiple information can be extracted based on grids, they failed to consider the effects of the road network. And the performance of these method highly relied on the quality of the raw data and the granularity of grids. Wang *et al.* [7] solved the problem by analogy with natural language processing. They mapped GPS points onto the road network and fed features of segments into LSTM to capture the sequence information. Though they managed to represent path precisely, their method to get the embedding of segments didn't fully utilize semantic information of segments in historical trajectories, and spatial features in Euclidean space was omitted. Gao *et al.* [8] represented the path as a sequence of segments and leveraged network embedding to represent segments. By mining the topological relationship between road segments, it predicted the travel time for unseen trip in the road network starting at any time. However, the embedding of segments overlooked variance along time and the path is only viewed in non-Euclidean space missing the information in Euclidean space.

C. Traffic Sequence Learning

With much sequential data obtained, there have been extensive research providing solutions about capturing useful information on sequence data. Previous work can be simply divided into two categories: convolutional neural network based methods, recurrent neural network based methods. Fu and Lee [19] and Yu *et al.* [20] utilized 1D-CNN to capture the temporal dependencies. Although this method can effectively capture local temporal patterns, it is not suitable for sequences with varying length which limited performance of this kind of method. Wang *et al.* [5] and Wang *et al.* [7] leveraged long short-term memory method to learn temporal dependency. And Zhang *et al.* [6] used bidirectional LSTM which combine a forward and backward LSTM's hidden output together. Some other work also leveraged other variants [21]–[23] of recurrent neural network to capture temporal dependencies in traffic sequences. In this work, we use the last hidden state of LSTM for estimation, which could aggregate information along the path.

TABLE I
MATHEMATICAL NOTATIONS

Notation	Comments
$\odot$	Hadamard product of two matrices
$[\cdot;\cdot]$	The concatenation operation of two vectors
$\mathcal{G}$	The road network
$\mathbb{V}$	The set of intersections
$\mathbb{E}$	The set of road segments
$\mathbf{h}_l$	The hidden state at l-th segment
$\mathbf{x}$	The vector of semantic representations
$\mathbf{W}, \mathbf{b}$	Trainable parameters

III. FORMALIZATION

The problem of travel time estimation is formalized formally in this section. Some notations used in this paper are shown in TABLE I.

A. Preliminaries

The travel time is defined as the time cost in driving from the origin to the destination. Our goal is to give an estimation of the travel time starting at a certain time.

Definition 1 (Road Network): The road network is a network which consists of road segments and intersections representing the driving road system of a city. In this paper, the road network is represented as $\mathcal{G} = (\mathbb{V}, \mathbb{E})$*, where* $\mathbb{E} = \{e_1, e_2, \cdots, e_N\}$ *is the set of* N *road segments and* $\mathbb{V} = \{v_1, v_2, \cdots, v_M\}$ *is the set of* M *intersections. Each intersection* v_m *contains a latitude and a longitude (*$v_m.lat$ *and* $v_m.lng$*). Each segment* $e_n = (v_{n,s}, v_{n,t}, e_n.type, e_n.length)$ *represents segment* e_n *exists from* $v_{n,s}$ *to* $v_{n,t}$ *with type* $e_n.type$ *and length* $e_n.length$*.*

Definition 2 (Trajectory): The set of trajectories is represented as $\mathbb{A} = \{a_1, a_2, \cdots, a_K\}$ *and a trajectory is represented as* $a_k = \{g_{k,1}, g_{k,2}, \cdots, g_{k,J}\}$ *which is a sequence of GPS sample points collected by vehicles. Furthermore, the* j*-th GPS sample point of* k*-th trajectory is represented as* $g_{k,j} = (lng_{k,j}, lat_{k,j}, t_{k,j})$ *which consists of a longitude, a latitude and a timestamp indicating the location of vehicles at time* $t_{k,j}$*.*

Definition 3 (Path): The path is a kind of data that doesn't contain time information and only has the information that describe how vehicles get to the destination from the origin. A path is represented as a sequence of segments on the road network $p = \{e_1, e_2, \cdots, e_L\}$*.*

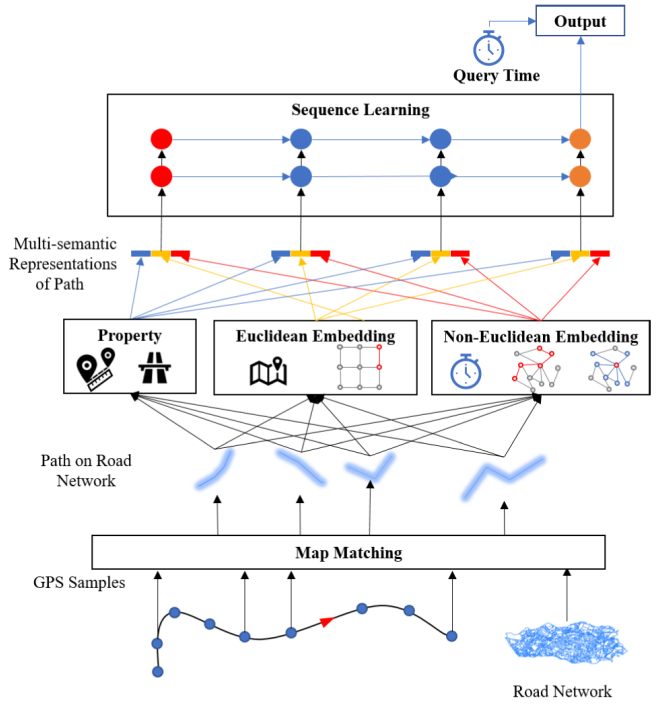


Fig. 2. The overall structure of the proposed model.

B. Problem Formalization

Travel Time Estimation: Given a road network $\mathcal{G}$ and a query $Q = (p, t)$ which represents a vehicle travel through the path p leaving at time t, travel time estimation aims to learn a function $f(\cdot)$ taking these information as input and output an estimation of the time cost $\hat{y}$ on this path:

$$\hat{y} = f(\mathcal{G}, Q). \tag{1}$$

IV. Methodology

Fig. 2 shows the overall structure of the proposed model. First, in the training phase, trajectories are transferred from sequences of GPS points to paths on road network through map-matching methods [24]. After that, each path can be viewed as a segment sequence or an intersection sequence, which are two different views to represent the same path. Based on this, multi-semantic representations are obtained to represent the path in multiple perspectives. Specifically, time-aware segment semantic representations are extracted to capture spatial features of path in non-Euclidean space. This part considers co-occurrence of segments which indicates relationships in non-Euclidean space, and the representations also adjust slightly according to time information. Moreover, semantic representations of intersections sequences are extracted from GPS coordinates, which contains spatial features of the path in Euclidean space. Then we combine these two types of semantic representations and semantic representations from properties together to construct the multi-semantic representations of a path. Finally, a sequence learning component is leveraged to aggregate the information along the path and an output layer is set to give estimation with aggregated hidden states.

A. Multi-Semantic Representations of the Path

1) Spatial Semantic Representations in Non-Euclidean Space: Since vehicles would travel along the road, the topology of the road network is crucial for estimating the travel time. Using one-hot encoding or the index number directly to represent segments is intuitive but limited to represent positions of segments in the road network. Here we extract semantic representations of segments in non-Euclidean space as a vector considering both historical co-occurrence of them and the time information.

First, we leverage a map-matching method so that every trajectory could be transferred to road network. GPS sample points are attached to segments, and continuous segments are then merged to remove the duplication. After that, the sequence is completed according to Shortest Path First algorithm which could recover segments and eliminate the impacts of the raw data quality. Second, the historical trajectories could indicate semantic relationships between segments including the connectivity and the priority of roads such as which pair of segments are more likely chosen by drivers at the same time. Given historical sequences of segments, the idea from SkipGram [25], [26] can help maximize the co-occurrence possibility of segments within a sliding window. Third, after training the segment embedding component, parameters of the hidden layer are used to embed segments to vectors. Through this, a static vector for each segment could be obtained. As the vector is extracted based on segments, it could represent the position of a segment in the road network. This process to get embeddings with parameter matrix outputted by Algorithm 1 can be described as a function:

$$\boldsymbol{x}_l^e = f^e(e_l), \tag{2}$$

where f^e is to transfer segment index to a vector and $\boldsymbol{x}_l^e$ represents the embedded vector of l-th segment.

The above steps obtain a static vector for each segment taking historical trajectories into consideration. It means the vector keeps the same regardless of the leaving time. However, it is known that the traffic status of the road network varies a lot across time. For instance, people are more likely to turn to segments leading to the entertainment area on weekends while turn to segments leading to the business area on weekdays. To address this deviation, we take a further step to generate time-aware semantic representations for segments. The representations are supposed to be based on segments' positions in the road network but vary slightly according to the human activities. To achieve this, three features of the leaving time are taken into consideration: time of day, day of week and day of year, which can capture the influence of rush hour, weekends and seasons respectively. For time of day, we split one day into 1440 slots, and each slot contains one minute. Therefore, we could get an index from 0 to 1439 to represent the leaving time of day in a query. After that, an embedding layer is utilized to embed the index to a vector $\boldsymbol{x}^d$. For day of week, we split one week into 7 slots and embed the index to a vector $\boldsymbol{x}^w$. And for day of year, we split one year into 365 slots and embed the index to a vector $\boldsymbol{x}^y$. Then, $\boldsymbol{x}^d, \boldsymbol{x}^w, \boldsymbol{x}^y$ are concatenated together to construct a vector $\boldsymbol{x}^t = [\boldsymbol{x}^d; \boldsymbol{x}^w; \boldsymbol{x}^y]$

for the time information. Finally, we leverage a residual block to obtain time-aware semantic representations of segments in non-Euclidean space:

$$\boldsymbol{x}_l^s = f^s([\boldsymbol{x}_l^e; \boldsymbol{x}^t]). \tag{3}$$

Algorithm 1 Segments Embedding

Input: $\mathcal{G}(\mathbb{V}, \mathbb{E})$: the latent road network
d: the embedding size
$\mathbb{A}$: the set of trajectories
w: the window size
E: the number of epochs
α: the learning rate
Output: the matrix of segment embeddings $\phi \in \mathbb{R}^{N \times d}$

Initialization: Sample ϕ from $U^{N \times d}$;
for $e = 1$ *to* E **do**
  foreach a *in* $\mathbb{A}$ **do**
    $attached\_segments = Match(\mathcal{G}, a)$;
    $Initializationofmerged\_segments$;
    foreach s *in* $attached\_segments$ **do**
      if s *is not* $merged\_segments.last()$ **then**
        $merged\_segments.append(s)$
    $p = FillBySPF(\mathcal{G}, merged\_segments)$;
    foreach v_j *in* p **do**
      foreach u *in* $p[j - w : j + w]$ **do**
        $J(\phi) = -logPr(u_k|\phi(v_j))$;
        $\phi = \phi - \alpha * \frac{\partial J(\phi)}{\partial \phi}$;
return ϕ;

2) Spatial Semantic Representations in Euclidean Space: As demonstrated in previous work [5], GPS data is highly related to the travel time, since GPS data can reflect spatial features of the path (e.g., how far and in which area vehicles travel). However, previous work simply used raw sampled GPS data so the performance highly relies on the quality of the raw GPS data. More specifically, both the sample frequency and the precision of GPS data may have influence on the performance of models. In this paper, GPS coordinates of intersections in matched path are utilized to address this issue. The sequence of intersections is a different view of the same path. Unlike semantic representations of segments, GPS coordinates of intersections reflect the path's spatial features in Euclidean space. Meanwhile, GPS coordinates of intersections from road network are fixed, which could reveal a more precise path than the sampled GPS data.

Segments are divided by intersections, so the sequence of intersections can be easily aligned with the sequence of segments by acting as sources or targets of segments. For each segment, a non-linear transformation is utilized to transfer GPS coordinates of its source intersection and target intersection to a vector $\boldsymbol{x}^g$:

$$\boldsymbol{x}_l^g = tanh(\boldsymbol{W}^g[v_{l,s}.lat; v_{l,s}.lng; v_{l,t}.lat; v_{l,t}.lng] + \boldsymbol{b}^g). \tag{4}$$

$\boldsymbol{x}^g$ contains the information of intersections' geographical positions which is spatial semantic representations of a path in Euclidean space.

3) Semantic Representations of Properties: After mapping GPS points to road segments, multiple properties information could also be extracted including properties of segments and properties of the path. Vehicles travel on a highway behave significantly different from those on a living street, which indicates that some properties of segments have a big impact on the travel time. In this paper, the length and the type of segments are taken into consideration. For the length, we directly use its value, and for the type of e_l, we embed it into a vector $\boldsymbol{x}_l^c$. Accumulating length is calculated by summing up the length of all segments before a segment. This feature can reflect the position of a segment in a path and the total length of a path. Finally, we concatenate all these information to form semantic representations of properties.

$$\boldsymbol{x}_l^p = [\boldsymbol{x}_l^c; e_l.length; \sum_{i \in E_f} e_i.length], \tag{5}$$

where E_f represents the set of segments before e_l in the query.

To fuse multiple semantic representations, we leverage two fully connected layers to learn representations of the path:

$$\boldsymbol{x}_l = \boldsymbol{W}_2 * ReLU(\boldsymbol{W}_1[\boldsymbol{x}_l^s; \boldsymbol{x}_l^g; \boldsymbol{x}_l^p] + \boldsymbol{b}_1) + \boldsymbol{b}_2, \tag{6}$$

And we finally construct a sequence of vectors containing multiple semantic representations of the path.

B. Sequence Learning Component

After getting the sequence of vectors, a sequence learning module is utilized to aggregate them. Considering the variation of sequences' length, LSTM [27] is applied as implementation of this part. At each step, multi-semantic representations of segments serve as the input and the gated mechanism is used to decide whether the information from the last step should be kept or forgotten.

$$\begin{aligned}
\boldsymbol{i}_l &= \sigma(\boldsymbol{W}_{ii}\boldsymbol{x}_l + \boldsymbol{W}_{hi}\boldsymbol{h}_{(l-1)} + \boldsymbol{b}_i).\\
\boldsymbol{f}_l &= \sigma(\boldsymbol{W}_{if}\boldsymbol{x}_l + \boldsymbol{W}_{hf}\boldsymbol{h}_{(l-1)} + \boldsymbol{b}_f).\\
\boldsymbol{g}_l &= \tanh(\boldsymbol{W}_{ig}\boldsymbol{x}_l + \boldsymbol{W}_{hg}\boldsymbol{h}_{(l-1)} + \boldsymbol{b}_g).\\
\boldsymbol{o}_l &= \sigma(\boldsymbol{W}_{io}\boldsymbol{x}_l + \boldsymbol{W}_{ho}\boldsymbol{h}_{(l-1)} + \boldsymbol{b}_o).\\
\boldsymbol{s}_l &= \boldsymbol{f}_l \odot \boldsymbol{s}_{(l-1)} + \boldsymbol{i}_l \odot \boldsymbol{g}_l.\\
\boldsymbol{h}_l &= \boldsymbol{o}_l \odot \tanh(\boldsymbol{s}_l).
\end{aligned} \tag{7}$$

As shown above, each hidden state is affected by previous hidden states which indicates that the l-th hidden state could capture information from $\boldsymbol{x}_1$ to $\boldsymbol{x}_l$. Therefore, the final hidden state could capture information of the whole path and is used to estimate the travel time.

C. Output Layer

The sequence learning component aggregates the multi-semantic information of a path into the final hidden states. Except for that, external elements can also affect the travel time. It is widely known that in rush hour people would take more time to finish a path than in free hour. Here we take $\boldsymbol{x}^t$ as time semantic representations of the query. Then, several fully connected layers with residual links [28] are used to estimate the travel time.

$$\hat{y} = f^o([\boldsymbol{x}^t; \boldsymbol{h}_L]). \tag{8}$$

We stack 3 residual fully connected layers, and the output of the final layer is the estimated time $\hat{y}$.

D. Optimization

Stochastic gradient descent strategy is used to train our model. And the objective function is MSE (mean square error):

$$\mathcal{L} = \frac{1}{K}\sum_{k=1}^{K}(\hat{y}_k - y_k)^2, \tag{9}$$

where K is the total number of the queries. The optimizer is Adaptive Moment Estimation [29] with 0.001 weightdecay and 0.0001 learning rate.

V. Experiment

This section will introduce experiments conducted on two large real-world trajectories datasets.

A. Datasets

The following part describes datasets used for the evaluation. For each dataset, we use map-matching methods to allocate trajectories to the road network and filter abnormal data outside the city road network or containing loops.

- *Porto Taxi:* The Porto dataset is available for an open challenge.[1] This dataset consists of trajectories generating by 442 taxis in Porto from Jul 1st, 2013 to Jun 30th, 2014. Each trajectory is recorded as a sequence of GPS sample points and each point contains a longitude and a latitude. The GPS points were sampled every 15 seconds, so the ground truth can be obtained using the leaving time and the number of points. Among these trajectories, there are some errors that are irrelevant to the task. For example, some trajectories has few GPS points while the vehicle travels a long way; this is intolerant because the true value of travel time comes from the number of GPS points, so we remove data with extreme ratio of raw GPS points sequence length and matched segments sequence length. The dataset consists of 426,023 trajectories and covers all of the city.
- *Chengdu Taxi:* The raw Chengdu dataset is in form of GPS points and each GPS sample point contains a longitude, a latitude, a timestamp, a load information and taxi identity number. This dataset is sampled by 14,864 taxis in Chengdu from Aug 3rd, 2014 to Aug 8th, 2014. Chengdu Taxi dataset is available for another online challenge.[2] Considering the situation that taxis may hang in a small area to find passengers when it's unloaded, only continuous loaded GPS sample points are extracted as trajectories. Using the same filtering strategy as Porto taxi dataset, we got a dataset containing 434,927 trajectories for the evaluation.

[1]Porto Taxi dataset is available at https://www.kaggle.com/crailtap/taxi-trajectory.

[2]Chengdu Taxi dataset is available at https://www.pkbigdata.com/

TABLE II
Details of Datasets

Data Source	Porto Taxi	Chengdu Taxi
Time from	2013/7/1	2014/8/3
Time to	2014/6/30	2014/8/8
Number of Trajectories	426,023	434,927
Number of Segments	11,342	23,965
Travel Time Mean	490.5749	538.4801
Travel Time Std	231.2591	408.8682

B. Baselines

The travel time estimation baselines include one statistic method TEMP_{rel}, three OD-based methods using random forest regression, XGBoost, ST-NN, one GPS-based method DeepTTE and two segment-based methods WDR, CTTE.

TEMP_{rel} [3]: A statistic method which aggregates historical travel time of adjacent trajectories. Considering time and space sparsity of datasets, TEMP with relative-time speed reference is compared here, and the distance to determine neighbors is adjusted to ensure that all trajectories in testing have corresponding neighbor trajectories.

Random Forest Regression [30]: Input of Random Forest Regression consists of the latitude and the longitude of OD pairs, day of week, day of year and time of day. The number of estimators is set to 100 and the objective function is mean square error.

XGBoost [31]: Input of XGBoost consists of the latitude and the longitude of OD pairs, day of week, day of year and time of day. The number of estimators is set to 100 and the objective function is mean square error.

ST-NN [1]: This model includes a distance module and a time module, each of which consists of multiple fully connected layers. It takes the latitude and the longitude of OD pairs and the time slot, which indicates time of day and day of week, as input to estimate the travel time and the distance of the whole path simultaneously.

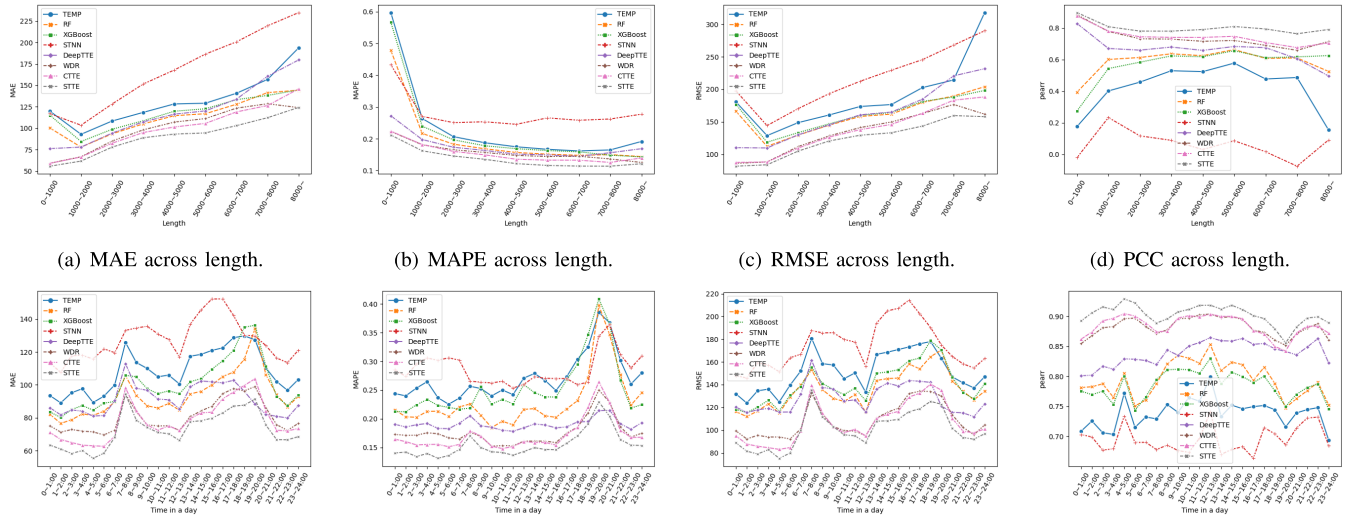
DeepTTE [5]: DeepTTE uses the sequence of GPS coordinates and load states to represent a path. Convolution layers are then applied to merge window-size points to representations of subpaths; LSTM is utilized to handle the sequence on the time axis.

WDR [7]: It is a method that jointly trains wide linear models, deep neural networks and recurrent neural networks to take full advantages of all three models. Extracted features include the number of segments, the length of path, the latitude and the longitude of OD pairs, the number of each type of segments and the number of each type of intersections.

CTTE [8]: It is a method that aggregates segments embeddings along the path. Due to the lack of the speed data and the driving behavior, only the part available in the dataset is kept.

C. Experiment Setups

For both datasets, we split them into three parts in the chronological order: training set, validation set and test set



(a) MAE across length. (b) MAPE across length. (c) RMSE across length. (d) PCC across length.

(e) MAE across time. (f) MAPE across time. (g) RMSE across time. (h) PCC across time.

Fig. 3. Detailed performance under different situations.

TABLE III
COMPARISON ON PORTO TAXI DATASET

Methods	MAE	MAPE	RMSE	PCC
$TEMP_{rel}$	109.4777	0.2708	154.0785	0.7461
RF	94.3659	0.2280	137.2945	0.8040
XGBoost	101.0125	0.2539	142.2613	0.7891
STNN	129.0142	0.2778	178.4647	0.6658
DeepTTE	92.2291	0.1902	129.9374	0.8471
WDR	80.9429	0.1743	110.1693	0.8789
CTTE	79.0731	0.1708	109.0702	0.8820
STTE	**73.5972**	**0.1640**	**100.8194**	**0.8998**

TABLE IV
COMPARISON ON CHENGDU TAXI DATASET

Methods	MAE	MAPE	RMSE	PCC
$TEMP_{rel}$	132.2510	0.3376	306.6909	0.7191
RF	129.9059	0.3599	286.7466	0.7606
XGBoost	134.1255	0.3686	293.0847	0.7467
STNN	168.5175	0.3760	333.2089	0.6594
DeepTTE	116.7453	**0.2181**	280.7787	0.7806
WDR	125.6295	0.2538	284.9564	0.7675
CTTE	133.3755	0.2661	290.5388	0.7655
STTE	**109.5367**	0.2399	**267.6178**	**0.7980**

that include 80%, 10%, 10% of the entire dataset respectively. Training set is used to train models and validation set is used to choose which model we evaluate on test set. In practice, some segments in test set never appear in training; we set their embeddings as vectors of all ones. For models using GPS based trajectories data, considering GPS points sampling are highly related to time, GPS points are resampled to make distances of two points are about 200m to 400m [5]. Experiments are conducted with Pytorch toolbox, scikit-learn toolbox and cuda 10.1 on 4 Geforce TitanXp GPUs.

We adopt MAE (mean absolute error), MAPE (mean absolute percentage error), RMSE (root mean square error) and PCC (Pearson correlation coefficient) to evaluate the performance of models. MAE is simply the average of error between the prediction and the true value. MAPE is sensitive to the error compared to the truth value and RMSE is sensitive to the extreme outliars. PCC reflects the correlation between the prediction and the truth value.

D. Comparison With Existing Models

As showed in Table III and Table IV, STTE outperforms all the other baselines including OD based methods, GPS based methods and segment based methods on these metrics which demonstrates the effectiveness of the proposed method. Besides, path based methods (DeepTTE, WDR, CTTE) outperform other baselines and the reason may be that more sufficient information is utilized than those OD based methods. As illustrated in Fig. 3, detailed performance on Porto Taxi Dataset with certain time and length shows that our proposed model could achieve better performance than other methods in most cases. The performance of OD based methods fluctuates as the length raises which may be caused by the loss of the information; this indicates that which way vehicles travel is important. And the performance of all the method drops in rush hour which may be caused by complex traffic states.

E. Ablation Study

A set of ablation experiments on Porto Taxi Dataset are also designed to demonstrate the effectiveness of each semantic representation. We design several variants by excluding certain part and replacing corresponding semantic representations to vectors of all ones. STTE w/o Time removes features of the departure time; STTE w/o SE removes semantic representations of the segments in non-Euclidean space; STTE

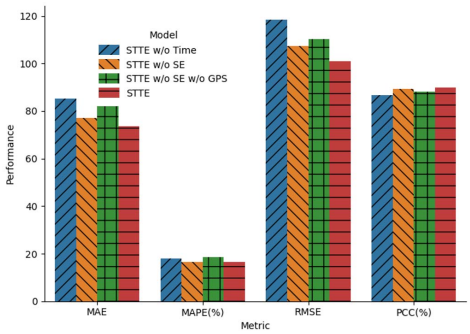


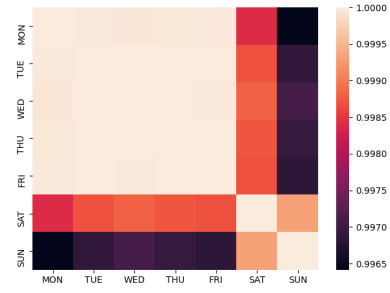
Fig. 4. Ablation study.

w/o SE w/o GPS removes spatial semantic representations of both intersections and segments. Other training setups of each experiments are kept the same.

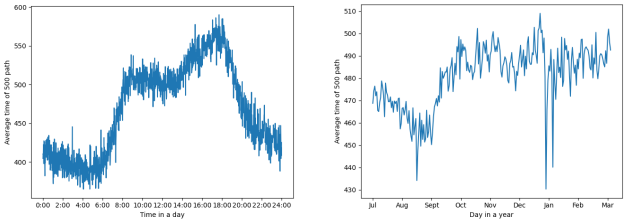
As showed in Fig. 4, the final model outperforms all the other variants without certain types of semantic representations. Among these, the performance of STTE w/o Time demonstrates the effectiveness of time semantic representations. This is reasonable since travel time would be longer in rush hour than free hour due to the potential congestion. A detailed analysis about the influence of time is provided in the next section. The improvement of time semantic representations is more than that of the segments. The reason of this phenomenon may be that semantic representations of intersections and segments alone may cover a part of spatial features. When we remove only one of them, the other one still contribute to the performance so it only causes a small gap. If we remove spatial semantic representations of both intersections and segments, the performance of the model would drop a lot. The performance of the final model is better than that of the model without semantic representations of segments, and the performance of model without semantic representations of segments is better than that of the model without features in both Euclidean space and non-Euclidean space; these phenomena demonstrate the effectiveness of these two spatial semantic representations.

F. Influence of the Leaving Time

We also give an illustration about the influence of the leaving time. We randomly pick 500 samples from Porto Taxi Dataset and change their leaving time of day from 0 to 1439 to construct a time-varied set. So as the week-varied set and the day-of-year-varied set. The illustration of results on these sets can be viewed in Fig. 5. For day of week, some segments take more time in weekdays while others take more time in weekends. It depends on the type of areas it pass, and we illustrate the similarity between these 500-dim result vectors. The results show a high correlation between weekdays and a low correlation between weekdays and weekends, which is reasonable considering people's activities. For time of day, the average time of 500 samples is calculated and it would cost more time in rush hour than free hour. For day of year, the average time of 500 samples is calculated and it costs more time in winter which may be caused by drivers' cautions and road conditions in winter. All of these results show that our



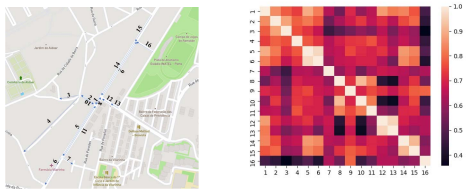
(a) Results with different day of week.



(b) Results with different time of day.

(c) Results with different day of year.

Fig. 5. Influence of the leaving time.



(a) Topology of the sampled road network.

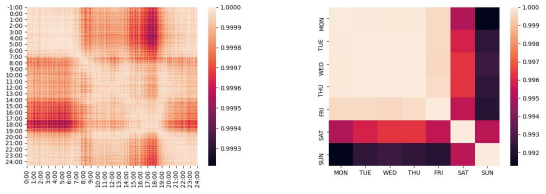
(b) Similarity between segments of sampled road network.

Fig. 6. Similarity of static segments embeddings.

methods could extract useful time semantic representations for estimating travel time. And the influence of time extracted by the model is correspond with people's activities.

G. Visualization of Segment Spatial Semantic Representations

Segments that are connected or likely to be traveled together are supposed to have similar representations. A subgraph of Porto road network is sampled to prove the effectiveness of segments embedding. As shown in Fig. 6(a), 16 segments are sampled and numbers on segments are the index of corresponding segments. The similarity of two segments' embeddings is defined as cosine similarity of these two vectors. The similarity between each pair from these segments is illustrated in Fig. 6(b). It is clearly shown that high similarity appears between segments sharing an intersection or having the same direction. Specifically, embeddings of segment 1-6 have the same direction which is more likely to be traveled together and have a high similarity; so are embeddings of segment 8-11. Moreover, embeddings of segment 5-6, segment 12-13 and segment 14-15 have a high similarity which indicates segments along a same road have stronger correlations; this is reasonable since most people tend to drive along road instead of turning too frequently. We also give an illustration about the effectiveness of segments' time-aware spatial semantic representations



(a) Similarity between segments' semantic representations with different time of day.

(b) Similarity between segments' semantic representations with different day of week.

Fig. 7. Similarity of segments' semantic representations.

by calculating similarity between semantic representations of segments at all time slots. As shown in Fig. 7(a), we change time of day to every minute in a day. The similarity indicates that the representations before dawn are highly related to those at night and are especially less related to those in the afternoon. And as shown in Fig. 7(b), we change day of week from Monday to Sunday. The similarity indicates that representations in weekdays are highly related to those in weekdays and are less related to those in weekends. These visualization reveals a pattern which fits our assumption and the human knowledge thus demonstrates the effectiveness of time-aware segment spatial semantic representations.

VI. Conclusion

Path representation is a rapidly developing field in recent year. In this paper, a novel multi-semantic representation model was proposed to estimate the travel time of a given path. A path was treated as a sequence of segments, and time-aware segments representation component captured spatial features of segments in non-Euclidean space. Moreover, the path was also viewed as sequence of intersections on the road network, and semantic representations based on this view provided spatial features in Euclidean space. Other semantic information including query time and properties of segments were also taken into consideration. Our multi-semantic representations method, which handles path's features from node view and edge view separately, could also be applied on other tasks to represent the path on general networks. Experiments have been conducted on two large real-world datasets from Porto and Chengdu, and the results demonstrated the effectiveness of our proposed model.

References

[1] I. Jindal, T. Z. Qin, X. Chen, M. Nokleby, and J. Ye, "A unified neural network approach for estimating travel time and distance for a taxi trip," 2017, *arXiv:1710.04350*. [Online]. Available: http://arxiv.org/abs/1710.04350

[2] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 1695–1704.

[3] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–22, Feb. 2019.

[4] R. Sevlian and R. Rajagopal, "Travel time estimation using floating car data," 2010, *arXiv:1012.4249*. [Online]. Available: http://arxiv.org/abs/1012.4249

[5] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? Estimating travel time based on deep neural networks," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[6] H. Zhang, H. Wu, W. Sun, and B. Zheng, "DeepTravel: A neural network based travel time estimation model with auxiliary supervision," 2018, *arXiv:1802.02147*. [Online]. Available: http://arxiv.org/abs/1802.02147

[7] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 858–866.

[8] R. Gao *et al.*, "Aggressive driving saves more time? Multi-task learning for customized travel time estimation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1689–1696.

[9] M. Asghari, T. Emrich, U. Demiryurek, and C. Shahabi, "Probabilistic estimation of link travel times in dynamic road networks," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2015, pp. 1–10.

[10] Z. Jia, C. Chen, B. Coifman, and P. Varaiya, "The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors," in *Proc. IEEE Intell. Transp. Syst.*, Aug. 2001, pp. 536–541.

[11] K. F. Petty *et al.*, "Accurate estimation of travel times from single-loop detectors," *Transp. Res. A, Policy Pract.*, vol. 32, no. 1, pp. 1–17, Jan. 1998.

[12] J. Tang, Y. Zou, J. Ash, S. Zhang, F. Liu, and Y. Wang, "Travel time estimation using freeway point detector data based on evolving fuzzy neural inference system," *PLoS ONE*, vol. 11, no. 2, Feb. 2016, Art. no. e0147263.

[13] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 5062–5068.

[14] P. Cintia, R. Trasarti, J. A. F. De Macedo, L. A. Cruz, and C. F. Costa, "A gravity model for speed estimation over road network," in *Proc. IEEE 14th Int. Conf. Mobile Data Manage.*, vol. 2, Jun. 2013, pp. 136–141.

[15] C. de Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real time floating car data," in *Proc. 11th Int. IEEE Conf. Intell. Transp. Syst.*, Oct. 2008, pp. 197–203.

[16] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 25–34.

[17] N. Zygouras, N. Panagiotou, Y. Li, D. Gunopulos, and L. Guibas, "HTTE: A hybrid technique for travel time estimation in sparse data environments," in *Proc. 27th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2019, pp. 99–108.

[18] K. Xiao, Z. Ye, L. Zhang, W. Zhou, Y. Ge, and Y. Deng, "Multi-user mobile sequential recommendation for route optimization," *ACM Trans. Knowl. Discovery From Data*, vol. 14, no. 5, pp. 1–28, Aug. 2020.

[19] T.-Y. Fu and W.-C. Lee, "DeepIST: Deep image-based spatio-temporal network for travel time estimation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 69–78.

[20] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*. [Online]. Available: http://arxiv.org/abs/1709.04875

[21] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 305–313.

[22] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," 2020, *arXiv:2012.08080*. [Online]. Available: http://arxiv.org/abs/2012.08080

[23] B. Du, X. Hu, L. Sun, J. Liu, Y. Qiao, and W. Lv, "Traffic demand prediction based on dynamic transition convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1237–1247, Feb. 2021.

[24] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden Markov model with precomputation," *Int. J. Geograph. Inf. Sci.*, vol. 32, no. 3, pp. 547–570, 2018, doi: 10.1080/13658816.2017.1400548.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: http://arxiv.org/abs/1301.3781

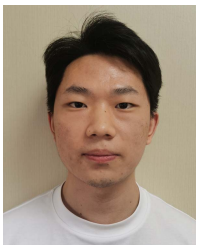
[26] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

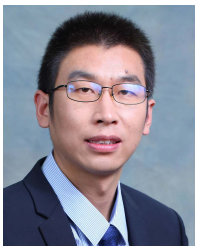
[30] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.



Liangzhe Han received the B.S. degree from the College of Software, Beihang University, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in computer science and engineering. His research interests include intelligent transportation systems, deep learning, and smart city technology.

Bowen Du (Member, IEEE) received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently a Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multi-source data fusion, and traffic data mining.

Jingjing Lin is currently a Senior Student at the School of Instrumentation and Optoelectronic Engineering, Beihang University, Beijing, China. Her research interests include travel behavior modeling, transportation simulation, traffic signal control, and spatiotemporal prediction and recommendation.



Leilei Sun (Member, IEEE) received the B.S. and M.S. degrees from the School of Control Theory and Control Engineering, Dalian University of Technology, in 2009 and 2012, respectively, and the Ph.D. degree from the Institute of Systems Engineering, Dalian University of Technology, in 2017. He was a Post-Doctoral Research Fellow with Tsinghua University from 2017 to 2019. He is currently an Assistant Professor with the State Key Laboratory of Software Development Environment, Beihang University, Beijing, China. His research interests include machine learning and data mining.

Xucheng Li received the B.Eng. degree in civil engineering and the Ph.D. degree in transportation from the University of Southampton, U.K. His research interests include deep learning, demand modeling, and intelligent mobility.

Yizhou Peng received the B.Eng. and M.Eng. degrees in communication and transportation engineering from Tongji University, China. He currently works as an Algorithm Engineer with Shenzhen Urban Transport Planning Center Company Ltd., Shenzhen, China.