How Fast You Will Drive? Predicting Speed of Customized Paths By Deep Neural Network

Hao Yang, *Graduate Student Member, IEEE*, Chenxi Liu, *Graduate Student Member, IEEE*, Meixin Zhu, Xuegang Ban, and Yinhai Wang, *Senior Member, IEEE*

***Abstract*— Customized path-based speed prediction is an eventful tool for congestion avoidance, route optimization and travel time prediction for navigation apps, cab-hailing companies and autonomous vehicles. Traditionally, the speed prediction algorithms are based on road segments and can only support several main roads. Path-based speed prediction is very challenging since the speed is always changing in different path locations and is jointly affected by lots of complicated factors. This article presents a novel deep learning framework for customized path-based speed prediction. A Path-based Speed Prediction Neural Network (PSPNN) is designed to achieve speed predictions for a given path and attributes information. A hierarchical Convolutional Neural Network (CNN) and deep Bidirectional Long Short-Term Memory (Bi-LSTM) structure for different kinds of feature extraction are applied for multiple levels: the path cell, sub-path and the whole path. The method narrows down the prediction unit from road segments to customized path cells (mean length: 59.52m) and achieves a mean absolute error (MAE) of 1.94 m/s and Mean Absolute Percentage Error (MAPE) of 18.14%, showing the potential of serving rigorous data-driven applications. So far, PSPNN is the first made-to-order path-based speed prediction algorithm and can help both travelers and managers to obtain large-scale bespoke paths speed information in advance.**

***Index Terms*— Speed prediction, path-based speed, deep learning, traffic information estimation.**

I. Introduction

With the rapid urbanization process worldwide, human beings suffer from the traffic congestion almost everyday. In recent years, with the terrific development of data collection and processing technologies [1], it has become a consensus to predict the trend and future status for both the transportation managers and travelers. Fortunately, traffic parameters prediction methods are being well investigated and are gradually put into real applications. Needless to say, speed prediction is one of the critical issues.

For traffic speed prediction, former research scopes could be divided into three levels: road segment [2], [3], corridor [4] and the network [5]–[8]. All these three prediction scopes are widely investigated in the past twenty years. For each prediction algorithm, the basic prediction unit is a road segment, and the value always represents the average velocity. These methods are useful for traffic managers and engineers, which can show and analyze the overall road network traffic status. However, for the individuals, they are more concerned about the traffic status of the paths where they will go. The former macroscopic level of speed prediction algorithms can not be well implemented into personal path planning. It is worth mentioning that many commercial navigation systems recommend the best route based on real-time segment estimation information [9], [10]. For example, based on real-time travel time and speed estimation, most commercial GPS navigation systems, i.e., Google Map, can realize the functions, including best travel path choice, real-time road conditions visualization, and best departure time recommendation. It is obvious that the "real-time best route" designed by the app at present may not be the real best route because there is a time lag from the users checking the app to reach the target road segment. The reality is that many people are all driven by the navigation software to the "smooth road segments at present" and suffer greater traffic congestion [10]. Therefore, a traveler-based individual level traffic information prediction algorithm should be proposed to address the problem.

With the development of data science and deep learning in the transportation field, researchers have paid more attention to customized-scaled prediction algorithms to help both managers and travelers obtain more useful prediction information [7]. However, customized path speed prediction is very challenging since the individual path-based speed prediction is affected by many complicated factors, including spatial and temporal factors (e.g., locations, peak hour/non-peak hour, departure time), human factors (e.g., drivers' behaviors, driving habits), internal conditions (e.g., vehicle performance) and external conditions (e.g., weather, road conditions). The traditional speed prediction methods can not perform well in such a challenging scenario. Therefore, a new algorithm is demanded, which not only can integrate more factors mentioned before, but also has the ability to serve for a new prediction scale. The previous speed prediction methods usually used the road segment as the basic prediction, while, in this article, the research team established a new deep neural architecture for customized path-based speed prediction to better serve both traffic managers and individual travelers.

In this new method, we propose 'path cell' as a new fundamental speed analysis and prediction unit, flexible and

Manuscript received September 18, 2018; revised July 6, 2019 and May 22, 2020; accepted September 30, 2020. Date of publication January 29, 2021; date of current version March 9, 2022. The Associate Editor for this article was J. Haddad. *(Corresponding author: Yinhai Wang.)*

The authors are with the Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: haoya@uw.edu; lcx2017@uw.edu; meixin92@uw.edu; banx@uw.edu; yinhai@uw.edu).

Digital Object Identifier 10.1109/TITS.2020.3031026

customized based on a given path in both spatial and temporal domains. Every path consists of customized path cells chain connecting end to end. The speed within a path cell is affected by the location, departure time, weather, road conditions, and so on. We proposed a deep neural network based on hierarchical Convolution Neural Network (CNN) and deep Long Short-Term Memory (LSTM) – Path-based Speed Prediction Neural Network (PSPNN). In the paper, we applied PSPNN on a real open dataset for tests, and the results prove the impressive ability of PSPNN on dealing with the challenging problem. Based on our research, we claim the contributions in the following five aspects:

1) Based on hierarchical CNN and deep bidirectional LSTM (Bi-LSTM) structures, a Path-based Speed Prediction Neural Network (PSPNN) is purposed to achieve speed predictions for a customized given path with starting points, endpoints, route, and attribute information.
2) Features extraction of spatial, temporal, environmental, and human-related factors is applied for multiple levels: the path cell level, sub-path level, and the whole path level.
3) Attributes information, including weather condition, road level, day of a week, trip departure time, and driver information are designed to integrate into the PSPNN based on embedding.
4) Through the PSPNN, the prediction unit can be narrowed down from fixed road segments to each personalized path cell (mean length of each cell: 59.52m).
5) At the micro-scaled path cell prediction level, PSPNN achieves a Mean Absolute Percentage Error (MAPE) of 18.14%, whose performance is significantly better than other cutting-edge deep learning benchmarks.

II. Literature Review

Typically, there are two categories of speed prediction modeling approaches: parametric and non-parametric [11]. Parametric methods are well-known as the statistical model-based approach, in which the parameters are predetermined based on theoretical assumptions. In conventional parametric methods, a series of mathematical models perform predictions dynamically, such as the kinematic wave model [2], cellular automaton model [12] and Boris Kerner's three-phase traffic theory [13]. Limited by the data acquisition techniques and large computation requirements, these model-based theories may not be fully applicable in practical scenarios.

Machine learning, especially deep learning, brings new energy to parametric methods. Recurrent Neural Network (RNN) is one of the popular neural networks in the transportation information prediction field. Compared with other neural networks, RNN can achieve good performance for sequence prediction. In the passing decades, many studies focused on RNN-based traffic information prediction and got good results. For example, [14] built up a road network speed prediction model based on custom RNN for network-scale speed prediction and got a promising result. Though RNN has a better performance than the traditional method, it still not good enough. Firstly, the basic RNN structures lack the ability to deal with the long-term of sequence memory. Secondly, because of the vanishing and exploding gradient problems, training RNN is difficult and time-consuming. Therefore, various RNN variants are proposed by researchers. Long Short-Term Memory (LSTM) neural network was designed to deal with these problems and replace the position of traditional RNN in transportation fields.

LSTM is one kind of special RNN, which is designed to learn long-term dependencies. Hochreiter and Schmidhuber proposed LSTM [15] in 1997 and refined by many researchers in the following decades. Therefore, at present, we can see various LSTM models designed for multiple purposes. In the transportation field, LSTM is one of the most popular models for traffic pattern prediction. Ma *et al.* [4] introduced the LSTM to travel speed prediction as the pioneer and got great results. In the following years, more and more studies [7], [16]–[24] have been done on the topic, and their results proved the performance of LSTM in travel speed prediction. With the long sequence processing ability, LSTM can examine the long-term dependencies, which lead to better results than RNN. However, a critical limitation of LSTM has been exposed: the dependencies are normally learned from chronologically arranged input data. As a result, only forward instead of backward dependencies can be extracted in the process. However, in the transportation field, both forward and backward dependencies are important for prediction. To capture both forward and backward dependency features, Bidirectional LSTM (Bi-LSTM) is introduced in the transportation field for travel pattern prediction [7], [17], [25], [26]. Wang *et al.* [7] applied multiple Bi-LSTM models in critical paths in the network and realized the multi-path network-scaled travel speed prediction. Cui *et al.* [25] proposed a stacked bidirectional and unidirectional LSTM (SBU-LSTM) neural network, combining LSTM and Bi-LSTM, for network-wide traffic speed prediction. The proposed model is capable of handling input data with missing values and is validated on both large-scale freeway and urban traffic networks in the Seattle area.

For traffic information prediction, both temporal dependency and spatial features are needed to be captured. For the spatial feature extraction, the Convolutional Neural Network (CNN) is widely used. Generally, CNN is usually applied for image and video analysis for its shared weighted architecture [27]–[30]. Therefore, taking advantage of the particular architecture, a lot of studies applied CNN to extract the spatial dependency in the road network in the transportation field. Researchers can extract spatial correlation features with the convolution operation and measure the influence of the given segment or intersection to the surrounding area by giving them various weights. For example, Ma *et al.* [31] realized the network-level traffic speed prediction based on CNN by using "learning traffic as image". Ke *et al.* [20] made lane-level traffic speed prediction considering the impact of traffic flow with CNN and got high accurate results.

Therefore, the combination of the structure of CNN and LSTM is empowered for temporal correlation and spatial feature extraction. Previous researches also proved that the idea is very encouraging and promising. Liu *et al.* [32] extracted the spatial-temporal information in urban area by Conv-LSTM for the short-term traffic flow prediction. Wang *et al.* [33] took advantage of CNN and LSTM for temporal-spatial information extraction for the end to end network scale trip travel time estimation, whose results greatly contribute to the route optimization and the navigation system. To the research team's best knowledge, the combination of spatial-temporal features, attribute information features and environmental features, and machine learning technology can better address the challenges for the customized path-based speed prediction problem.

III. Model Preliminary

A. Definition

Definition 1 Historical Trajectory (H_i): a historical trajectory H_i is a sequence of continuous historical GPS points generated by vehicle i with a fixed length of time gap, i.e., $H_i = \{h_1, h_2 \ldots h_{m-1}, h_m\}$. Each point record of h_m includes latitude, longitude and timestamp. In this research, the trajectory is the basic information for our model.

Definition 2 Path (P_i): a passable pathway with given origination, destination, and route on the map. In this research, P_i is generated by users, and it is the fundamental entity for researchers to predict the speed sequence information. Each P_i includes multiple path cells and sub-paths.

Definition 3 Path cell ($C_{P_i}^n$): the customized isometric sections of each path (P_i). Path cell is the basic path speed prediction unit. N represents the cell number and belongs to the P_i $(0 < n < j)$ and $Dis_{C_{P_i}}$ is the distance of each path cell.

Definition 4 Sub-path ($S_{P_i}^N$): sub-path is the combination of k adjacent path cells. Sub-path is used to reflect and summarize and the neighboring path cells connections and relationships. The sub-path length equals to $k \cdot Dis_{C_{P_i}}$.

B. Research Objective

Randomly given a customized path (path) (P_i) on a map with origination, destination, path starting time, route and attributes information, then predict the speed ($V_{C_{P_i}^n}$) belonging to each path cell ($C_{P_i}^n$).

C. Data Pre-Processing

In this research, path information and the path cell information are obtained from the trajectory data. The historical trajectory (H_i) GPS points are the original records holding the spatial-temporal information. Based on the definitions, after one historical trajectory obtained, the first step is to calculate the length (Dis_{P_i}) of H_i. Based on the Dis_{P_i}, the second step is to divide H_i into n GPS points clusters. Each cluster covers the same distance, which is treated as path cell length. The third step is to sample each cell sequence cluster and make the total records equal to j (if the sequence is not long enough, add 0 to each cell record).[1] Using the sampled GPS points in each isometric cluster represent each path cell ($C_{P_i}^n$). The fourth step is to add the cell information (cell length $Dis_{C_{P_i}}$) and the geographic information (road level) to each point. An example of the overall data pre-processing process is shown in Fig. 1.

After the data pre-processing, we obtained the path P_i, consists of j records representing for n path cells, and each data record in the path cells $p_{C_{P_i}^n}^j$ includes longitude, latitude, the customized cell length $Dis_{C_{P_i}}$, and the road level (RL).

$$p_{C_{P_i}^n}^j = (Long, Lat, Dis_{C_{P_i}}, RL) \tag{1}$$

D. Attributes

In this research, attributes information is an important input of the model, including: weather ($weatherID$), week date ($weekID$) and driver's info ($driverID$). These attributes are presented as attributes set, represented as A, defined in equation (4). Here, weatherID represents weather conditions, such as rain, snow, or clear, and weekID indicates the week day from Monday to Sunday. DriverID is used to distinguish the drivers.

$$A = (weatherID, weekID, driverID) \tag{2}$$

Besides, some path attributes are summarized including:

- starting time: in the unit of second, over a range of (0, 86400), as a day is divided into 86400 intervals in the paper.
- current average speed of different road levels (update every one minute).

IV. Model Architecture

The architecture of PSPNN is shown in Fig. 2. It essentially consists of two modules: the Feature Extraction Module (FEM) and the Memory Module (MM). The FEM is responsible for extracting and converting features of a particular path. There are four parts in this module: the Cell-Conv layer, the Sub-path Conv layer, the Path-Conv layer and the attributes embedding. The MM is used to memorize and summarize connectivity of cells. The basic structure of this module is deep Bi-LSTM. Then the outputs of the two modules will be input in four fully-connected layers to map the vector matrix into prediction speed sequence.

A. Feature Extraction Module (FEM)

1) Cell-Conv Layer: The Cell Convolution layer is used to capture features including in a sequence of path cell records. It is responsible for converting the raw sequence to a series

[1]Generally, GPS data is recorded by a relatively fixed time gap. Here, we re-sample the data set since

1) To avoid network learning error (e.g., just predicting the speed based on counting the GPS record numbers);
2) To save computing resources and increase training speed (some of the path records including more than 3k GPS records);
3) To improve the model applicability to both long and short customized path speed prediction.

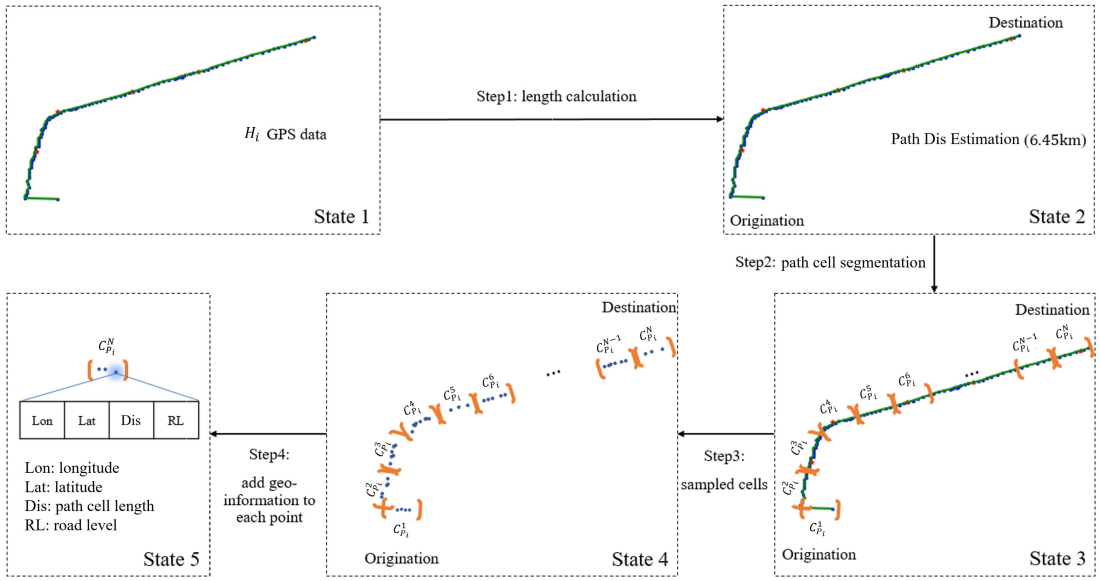


Fig. 1. Data pre-processing, using one real path as an example (Vehicle ID:00f8c1861b883d4a9bde0babcbbe2f96).

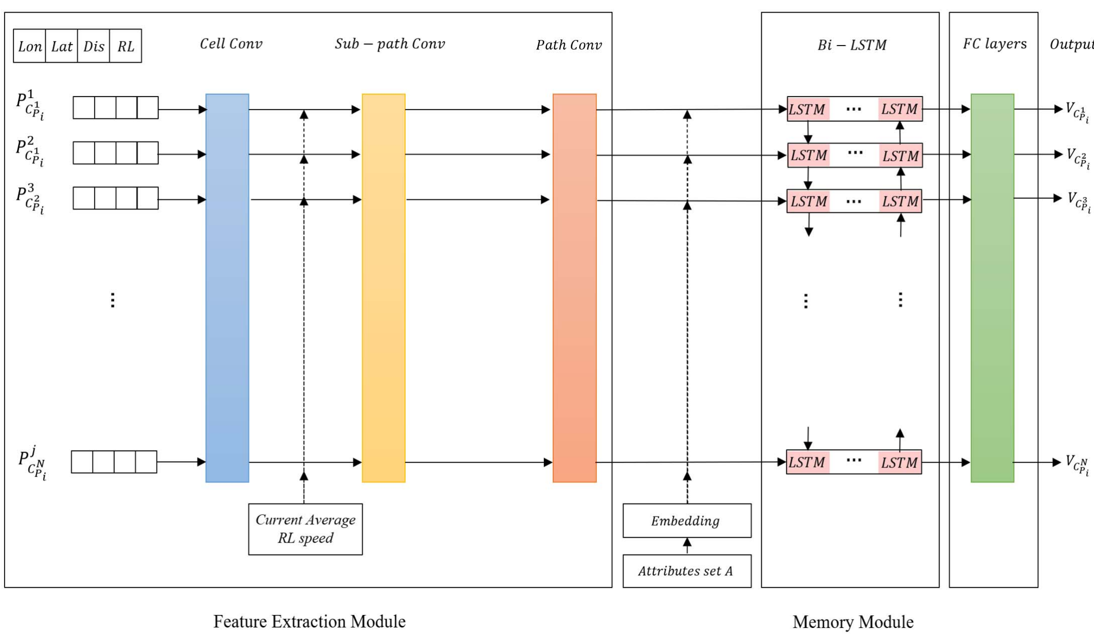


Fig. 2. Path-based Speed Prediction Neural-Network (PSPNN) Architecture.

of feature vectors. Inspired by the Geo-conv Layer of the DeepTTE model proposed in Wang *et al.* [33], the Cell-Conv Layer is used to capture the microscopic spatial correlation among consecutive trajectory cell records. The detailed architecture of Cell-conv is shown in Fig. 3. Here, a non-linear mapping is applied to integrate the three parts into the Cell-Conv layer, a 1-D convolution layer and concatenate. Before the convolution calculation, the authors mapping the input vector $p^j_{C^n_{P_i}}$ from R^4 to R^{20}. This step is used to enlarge the records dimension for better records feature extraction. Thus, the input sequence of the convolution layer can be treated as 20 channels. The mathematical approach of the Cell-Conv layer is in the following:

$$f_{p^j_{C^n_{P_i}}} = tanh(W_{geo}[long \circ lat] \circ W_{dis}[Dis] \circ W_{rl}[RL]) \tag{3}$$

$$f^{cell}_{p^j_{C^n_{P_i}}} = \sigma_{cnn} \cdot (W_{conv} * f_{p^j_{C^n_{P_i}}:p^{j+k_c-1}_{C^n_{P_i}}} + \varepsilon) \tag{4}$$

In the equation (3), the $f_{p^j_{C^n_{P_i}}}$ means the output of the non-linear mapping of output. The W_{geo} W_{dis} W_{rl} are learnable weight matrix. The non-linear mapping method used *tanh* function. In the equation (4), $*$ represents the convolution

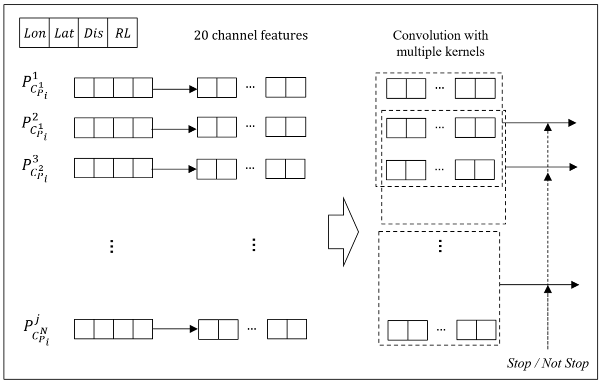


Fig. 3. The structure of the Cell-Conv Layer.

calculation with kernel size k. ε is the bias item. $p^{j}_{C^{n}_{P_i}} : p^{j+k_c-1}_{C^{n}_{P_i}}$ is the input sequence of a path from records j to $j+k_c-1$. The k_c is the kernel size of the convolution layer. σ_{cnn} represents the corresponding activation function of CNN. The output of the convolution layer $f^{cell}_{p^{j}_{C^{n}_{P_i}}}$ shows as equation (4).

2) Sub Path Conv Layer: Generally, a vehicle's speed on a certain road section is always affected by the adjacent road sections. Therefore, it is necessary to use the adjacent road segments' traffic information to assist the algorithm in predicting in the target cell. Here, a Sub-path convolution layer is designed to capture the features among adjacent cells. Here, before transmitting the cell convolution layer's output into the Sub-path convolution layer, the road network average speed of different road levels AS^{j}_{rl} is concatenated to each path cell at the trip time. This is aimed to help the network capture the current traffic status better. And the output of sub-path convolutional layer is $f^{sub}_{p^{j}_{C^{n}_{P_i}}}$:

$$f^{insub}_{p^{j}_{C^{n}_{P_i}}} = (f^{cell}_{p^{j}_{C^{n}_{P_i}}} \circ AS^{j}_{rl}) \tag{5}$$

$$f^{sub}_{p^{j}_{C^{n}_{P_i}}} = \sigma_{cnn} \cdot (W_{conv} * f^{insub}_{p^{j:j+k_s-1}_{C^{n}_{P_i}}} + \varepsilon) \tag{6}$$

3) Path Convolution Layer: The path convolutional layer is used to capture a high level of path features across several cells. Here, we still use a 1-D convolution layer to finish the feature extraction. The math equation of path convolutional layer is in equation (7):

$$f^{path}_{p^{j}_{C^{n}_{P_i}}} = \sigma_{cnn} \cdot (W_{conv} * f^{sub}_{p^{j:j+k_p-1}_{C^{n}_{P_i}}} + \varepsilon) \tag{7}$$

4) Attributes Set and Embedding: The traffic system can be treated as a subsystem of the modern social system. The parameters will be affected by many general factors. Travel patterns on holidays and weekends also vary greatly from working days. Furthermore, the travel speed can be different in different weather conditions. To address this, the model includes and integrates the attributes set into the path speed prediction. Here, the attributes set includes weatherID (rainy, snowy, sunny, etc.), weekID (from Monday to Sunday), departure time ID and the driverID.

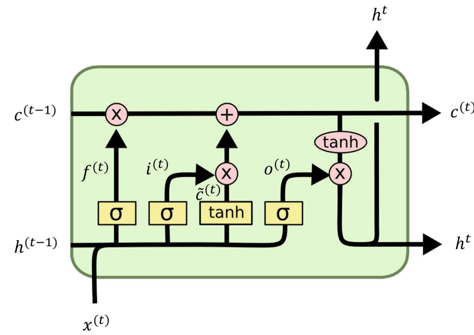


Fig. 4. The structure of the LSTM neural [15].

However, the values of the attributes are always categorical values and not satisfied with the neural network input format. A learnable procedure is necessary to address the problem since the impact of the attributes to the output is always complicated and associated. Inspired by feature learning techniques in natural language processing (NLP), mapping the words or phrases from the vocabulary to vectors of real numbers, embedding becomes a bridge to connect these discrete values to a vector dimension. In the framework, we adopted the low dimension embedding method proposed by [34] to transform categorical factors into a neural network input sequence. Using $E(A)$ represents the four attributes vectors. The overall output of the memory module is:

$$f^{FE}_{p^{j}_{C^{n}_{P_i}}} = f^{path}_{p^{j}_{C^{n}_{P_i}}} \circ E(A) \tag{8}$$

B. Memory Module (MM)

The change in velocity between different traffic cells can be interpreted as a change in the spatial and temporal domain. After the feature extraction module extracting and summarizing the spatial features, a module with time memory properties is needed. LSTM is a well-known network to capture the temporal dependencies among these cells. The LSTM is a special form of RNN, which is being capable of learning long-distance dependencies and handling long-term memory information. Generally, each LSTM neural contains three gates, which are input gate $i^{(t)}$, output gate $o^{(t)}$ and forget gate $f^{(t)}$ inside a neural. Each gate is controlled by their own weight $w^{(t)}$ and the previous neural output $h^{(t-1)}$. Also, the memory cascading process can be divided into two parts: new memory generation $\tilde{c}^{(t)}$ and final memory generation $c^{(t)}$. After the last memory is generated, the new hidden state $h^{(t)}$ is raised by the control of output gate $o^{(t)}$. $o^{(t)}$ makes the assessment regarding what parts of the memory need to be shown in the $h^{(t)}$. Fig. 4 shows the detailed structure of LSTM neuron and the mathematical formulations of (9)-(14) show the LSTM units working procedures in the following [15]:

$$i^{(t)} = \sigma(W^{(i)}x^{(t)} + U^{(i)}h^{(t-1)}) \tag{9}$$

$$f^{(t)} = \sigma(W^{(f)}x^{(t)} + U^{(f)}h^{(t-1)}) \tag{10}$$

$$o^{(t)} = \sigma(W^{(o)}x^{(t)} + U^{(o)}h^{(t-1)}) \tag{11}$$

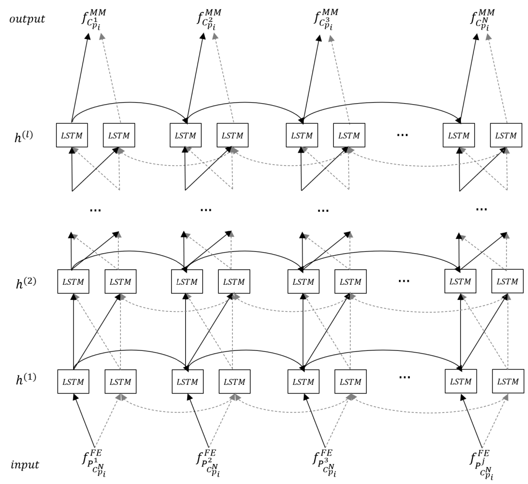


Fig. 5. Deep bidirectional LSTM neural network architecture.

$$\widetilde{c}^{(t)} = tanh(W^{(c)}x^{(t)} + U^{(c)}h^{(t-1)}) \tag{12}$$
$$c^{(t)} = f^{(t)} \circ c^{(t-1)} + i^{(t)} \circ \widetilde{c}^{(t)} \tag{13}$$
$$h^{(t)} = o^{(t)} \circ tanh(c^{(t)}) \tag{14}$$

To capture the time dependency better, PSPNN uses deep bidirectional LSTM as the memory module. Each training sequence is passed the LSTM neural from two directions, one forward and another backward. The mathematical formulations of deep bidirectional LSTM are showing below:

$$\overrightarrow{h}_t^{(i)} = f(\overrightarrow{W}^{(i)}\ h_t^{(i-1)} + \overrightarrow{V}^{(i)}\ h_{t-1}^{(i)} + \overrightarrow{\varepsilon}^{(i)}) \tag{15}$$
$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)}\ h_t^{(i-1)} + \overleftarrow{V}^{(i)}\ h_{t+1}^{(i)} + \overleftarrow{\varepsilon}^{(i)}) \tag{16}$$
$$\hat{y^{(t)}} = g(\mathcal{U}[\overrightarrow{h}_t^{(i)}; \overleftarrow{h}_t^{(i)}] + \hat{\varepsilon}) \tag{17}$$

The Fig. 5 and the equation (15)-(17) show the bidirectional hidden layer architecture parameters updating process. Generally, the bidirectional LSTM consists of two parts, the forward sequence and backward sequence. During the calculation process, the network will compute the forward hidden sequence $\overrightarrow{h}_t^{(i)}$ and the backward hidden sequence $\overleftarrow{h}_t^{(i)}$. Finally, the output integrates both forward and backward hidden features and summarizes into one sequence as output.

To narrow down the dimension of memory module output and obtain the cell speed sequence, multiple Fully Connected (FC) layers are used here. These layers' task is to map the high dimension feature sequence to a specific scale of value. Based on the mapping results, the predicted value of each path cell $V_{C_{P_i}^n}$ can be obtained.

V. Experiment

A. Environment Description

The PSPNN was implemented with PyTorch 0.3.1. The work station for training was equipped with a GPU (NVIDIA TITAN Xp) and the CPU is Intel Core i7 8700. The operating system is Linux Ubuntu 16.04.

B. Data Description

Overall description of the research dataset: Didi GIYA data from Chengdu, China. The original data size was about 132 GB and contained more than two billion trajectory records from November 1st, 2016 to November 30th, 2016. After the data cleaning, the effective path travel time ranges between 3.25 and 59.97 minutes. The trip distance is ranging from 0.12 km to 35.74 km. The GPS points recording time gap is 2-4 seconds.

Also, we match the trajectory with the Chengdu city road network to extract the road level information. Here, researchers divided the road into five different RL as input, including: freeway, arterial streets, sub-arterial streets, collector's streets and local streets.

C. Parameter Setting

The parameters in the PSPNN experiment are as follows:

- The customized cell amount n in PSPNN for paths is fixed as 128. The customized point records amount j in PSPNN for paths is fixed as 256. The average length of each cell is 59.52m.
- In the Cell-Conv layer, the kernel size k_c is fixed as 3. The number of filters is set as 128. The activation function in Eq (4) σ_{cnn} is Exponential Linear Unit (ELU) [35] function, which can converge cost to zero faster and produce more accurate results. Also, ELU has an extra alpha constant which should be positive number.
- In the sub-path convolution layer, we fixed the kernel size as 2. Therefore, the sub-path is composed of three adjacent path cell clusters and represent the transmission and connection among the two neighboring cells clusters. The number of filters of the sub-path convolution is set as 128. The activation function σ_{cnn} is ELU function.
- In the path convolution layer, we fixed the kernel size as 3. The number of filters of the sub-path convolution is set as 96. The activation function σ_{cnn} is ELU function.
- The size of the embedding vector for each attribute is settled in the following: weatherID mapping into R^3, weekID mapping into R^3, departure time ID mapping into R^{16} and the driverID mapping into R^{10}. The total dimension size of $E(A)$ is R^{32}.
- In the memory module, the number of hidden neural in the bidirectional LSTM is fixed as 256. Here, three hidden layers are used in the memory module. The activation function in Eq (9-11) σ_{rnn} is $tanh$ function. The mathematical expression of $tanh$ is $tanh(x) = e^x - e^{-x}/e^x + e^{-x}$
- In prediction module, the number of fully connected layers is fixed as 4. The four layers down sample to the 128 dimension vector to represent as the predicted cell speed ($V_{C_{P_i}^n}$).

D. Loss Function

In PSPNN, the researchers use the mean absolute percentage error (MAPE) to train PSPNN model. And then using the mean absolute error (MAE), root mean squared error (RMSE) and MAPE together to evaluated the model. The loss function and

TABLE I
PERFORMANCE COMPARISON

Method	MAE (m/s)	RMSE (m/s)	MAPE
AVG	4.37 (15.44km/h)	6.32 (25.76km/h)	87.98%
Deep RNN	3.32 (11.95km/h)	5.14 (18.50km/h)	52.78%
Deep LSTM	3.13 (11.27km/h)	4.77 (17.17km/h)	47.89%
Deep Bi-LSTM	3.07 (11.05km/h)	4.58 (17.49km/h)	46.02%
Conv-LSTM	2.89 (10.40km/h)	4.22 (15.19km/h)	37.95%
PSPNN (no embed)	2.13 (7.67km/h)	3.36 (11.38km/h)	19.96%
PSPNN	**1.94 (6.98km/h)**	**2.98 (10.73km/h)**	**18.14%**

evaluation methods for the cell speed prediction are defined in the following equations:

$$\mathrm{MAE} = \frac{1}{N} * \sum_{n=1}^{N} |V_{C_{P_i}^n} - \hat{V}_{C_{P_i}^n}| \tag{18}$$

$$\mathrm{MAPE} = \frac{1}{N} * \sum_{n=1}^{N} |\frac{V_{C_{P_i}^n} - \hat{V}_{C_{P_i}^n}}{V_{C_{P_i}^n} - \varepsilon}| * 100\% \tag{19}$$

$$\mathrm{RMSE} = \sqrt[2]{\frac{1}{N} * \sum_{C=1}^{N} (V_{C_{P_i}^n} - \hat{V}_{C_{P_i}^n})^2} \tag{20}$$

In the equations (18)-(20), the $V_{C_{P_i}^n}$ represents the predicted speed value of each path cell. The $\hat{V}_{C_{P_i}^n}$ represents the ground truth speed of each path cell. n is the cell number. In equation (19), ε in MAPE is used to prevent the 0-denominator appearance and ε is small enough compared with the average road network speed.

VI. Performance Evaluation and Comparison

The training process is reasonable. The MAPE training curve dropped slowly and converged to a limit at 18.14%. During the evaluation, the MAE drops to 1.94 m/s (6.98 km/h). Additionally, to better show the performance of PSPNN, the paper also comparing with a traditional method (AVG) and other cutting-edge deep neural network architecture, including deep RNN, deep LSTM, deep Bi-LSTM, deep Conv-LSTM, and PSPNN without embedding. The results are summarized in Table I. Also, the parameters of each deep neural network using in the comparison are shown in the following items.

- AVG [36]: Calculating average speed for each path is the most traditional method for path-based speed prediction. In this method, researchers calculated the average speed of each path and used the average speed as the predicated cell speed. Then the paper estimates the errors based on real path cell speed and the average speed.
- Deep RNN [5]: RNN can use internal memory units to process arbitrary sequences of inputs, and thus grants the RNN the capability of learning temporal sequence. In the comparing process, the authors used a primary recurrent neural network (RNN) to predict cell speed and compared it with the real path cell speed. Here, we set the neural number of RNN as 256 and using three hidden layers.
- Deep LSTM [4]: For comparison, the authors used deep LSTM neural network to predict the cell speed directly. Here, we set the number of hidden units as 256 and in three hidden layers. Then based on the LSTM output, the paper estimates the errors based on real path cell speed and the predicted cell speed.
- Deep Bi-LSTM [6]: For comparison, the authors also used deep Bi-LSTM neural network to predict the cell speed directly. Here, we set the number of Bi-LSTM hidden unites as 256 and in three hidden layers. Use Bi-LSTM neural network to predict the cell speed directly and then estimate the errors.
- Deep Conv-LSTM [37]: Combined both CNN and LSTM as a new neural network into Convolutional LSTM (Conv-LSTM) is tested here. Comparing with traditional LSTM, convolutional layers are used for feature extraction, which can extract not only the temporal correlation but also the spatial correlation. Here, a 1D-CNN and a deep LSTM network is used to capture both temporal and spatial features. For the 1D convolution layer, we set the kernel size equals to 3 and the number of filters is 128. Then, passed a three hidden layers deep LSTM with 256 hidden units.
- PSPNN without embedding: in this research, the authors implemented the embedding methods into the path based on speed prediction and try to integrate features such as driver behavior, day of week, and the weather impacts. To see the actual influence, we trained and evaluated a PSPNN without the attributes of information embedding.

From the comparison of table I, we can see that the PSPNN prediction accuracy is significantly better than other methods. Comparing with the traditional deep RNN and LSTM, the feature extraction module do extract more useful information. Also, we found that the convolution layer do help the spatial features extraction since the Conv-LSTM performs around 9.94% comparing with only using the same hidden layers of deep LSTM network. More detail analysis about the PSPNN components can be found in the next section.

VII. Result Discussion and Analysis

A. Feature Extraction Module

1) Integration of Attributes Information: In real life, people's travel is always affected by the many external factors such as weather, departure time, day of the week and etc. How these factors impact the prediction result? Here, the research team did an evaluation with and without the attributes set, and we found that the full attributions information with embedding methods using in the PSPNN can improve the overall prediction MAPE of 1.82%. Among the four attributes, the weekID shows the most significant contribution to the result with a 1.02% improvement. Then, the weatherID and the timeID provide with 0.91% and 0.73% error decrease for the MAPE. The influence of the driverID is neglectable, which shows about 0.05%-0.08% contribution to the MAPE accuracy. The result shows that the driving habit still not be fully found out and integrated into the neural network. A more detailed attributes to represent driver's behavior is needed in future research.

2) Cell Convolution Layer: For the cell convolution layer, we tested multiple parameter combinations. For the kernel size k_c, we tried different values $k_c = 2, k_c = 4, k_c = 5$ and

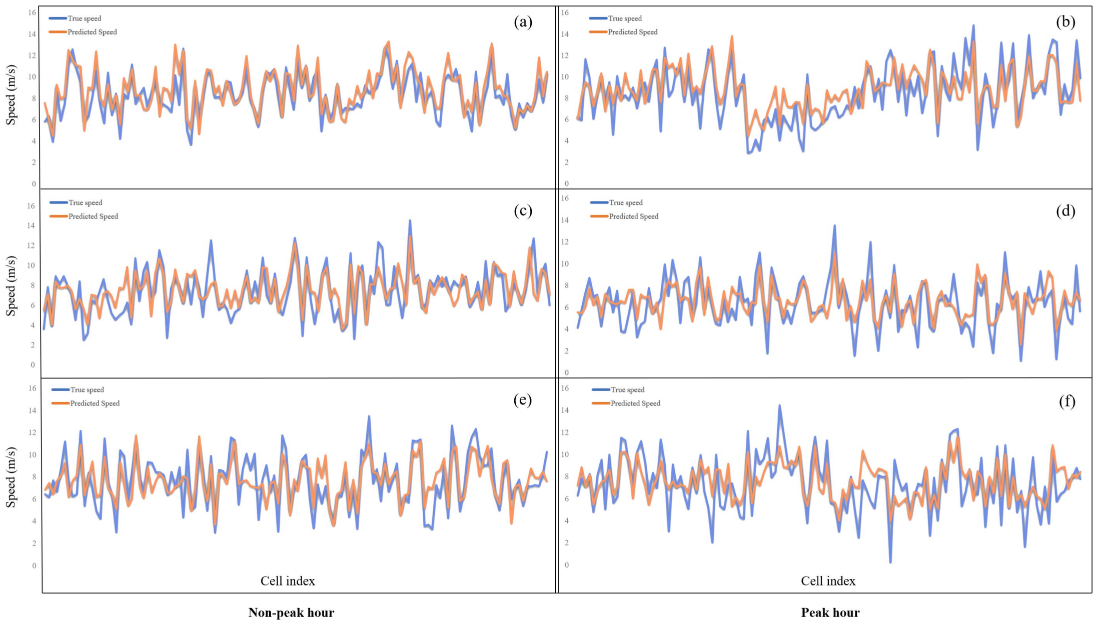


Fig. 6. Comparison of predicted path cell speed and ground truth of six random paths.

the prediction MAPE were worse than $k_c = 3$. Because the cell convolution layer is mainly used to extract the spatial dependency in the small cell clusters. It is obvious that the speed in a path cell is highly dependent on the former and latter cells. Also, the vehicle 's speed change will show an influence on the neighborhood cells. The same result of the kernel size attempt was mentioned in the [33] Geo-Conv layer.

3) Sub-Path and Path Convolution Layer: For the sub-path convolution layer, we finally set the kernel size $k_s = 2$. Here, we also tested several different kernel sizes that $k_s = 3, k_s = 4$, and we obtained different MAPE results as 19.31% and 20.23%. The possible reason leading the result is that the sub-path convolution layer is mainly responsible for capturing the higher dimension features, especially for the connections to each cell clusters. The $k_s = 2$ potentially shows that the speed change in a path cell is closely to the neighborhood cells status. Also, for the path convolution layer, we set the kernel size as 3. We also tried the different kernel size $k_p = 2, k_p = 4$ and $k_p = 5$. We found that the MAPE are 21.07%, 19.85%, 19.98%.

We also tried different number of filters in the three different convolution layers. After considering the prediction performance and network efficiency, we finally decided the number of filters as showing in the section VI.

B. Memory Module

In the PSPNN, the memory module is a significant part of capturing the temporal dependency for the speed changes among different cells. The paper tries different models and different hidden layers, and finally decides to use Bi-LSTM since its performance is the best. We can infer this result from the table I from the comparison of different types of RNN. For the hidden layers, we tried the $l = 2$ and $l = 4$. We found that when $l = 2$, the MAPE result (19.04%) is pretty close to the best one. When the $l = 4$, the MAPE is 18.09%, whose improvement can be neglected. However, increasing one Bi-LSTM layer with 256 hidden neural will lead into huge parameter growth. So, the research team finally decided to use three hidden layers Bi-LSTM as the memory module.

C. Path-Based Speed Prediction Case Analysis

It can be seen from the plots that the PSPNN has captured the changes in the temporal and spatial aspects of the entire path speed in fig 6. Here, we randomly selected six vehicles and six testing paths, of which three in non-peak hours and three in the peak hours. The x-axis indicates the cell index (from 0-127) and the y-axis is the speed value (m/s). The blue line shows the real speed captured by the historical trajectory records and the orange line shows the predicted cell speed of the path. Form Fig. 6, we can see that the overall forecasting trend is very close to the actual speed of the path, especially when the driving speed is close to the mean road network speed. The success of the PSPNN shows that the combination of PSPNN has an outstanding performance in modeling complicated spatial-temporal features.

However, several findings still worth for an open-minded discussion. The first and foremost is that the cells, usually with a higher speed is easy to predict comparing with the low speed cells (under 4m/s). The details can be found from Fig. 6. This results could be explained by the following points. Firstly, in the road network, the stable high travel speed always indicates the high spatial dependency. Generally, in the situation, the road surface and the transportation infrastructure are both in good conditions. Also, the speed limit and the level of services of the road segments are higher. These obvious spatial

TABLE II

ANALYSIS OF DIFFERENT DEPARTURE TIME AND ERROR

Departure Time	MAE (m/s)	RMSE (m/s)	MAPE (%)
5:00-7:00	1.85±0.05	2.89±0.05	17.19±0.2
7:00-9:00	2.31±0.3	3.32±0.2	21.13±0.4
9:00-11:00	1.93±0.1	2.92±0.1	18.19 ±0.25
11:00-13:00	1.91±0.05	2.93±0.05	19.06 ±0.2
13:00-15:00	1.89±0.05	2.95±0.05	17.76 ±0.2
15:00-17:00	2.01±0.15	3.02±0.1	20.01 ±0.25
17:00-19:00	2.55±0.4	3.45±0.5	25.31 ±0.5
19:00-21:00	1.90±0.1	3.01±0.04	17.28 ±0.2
21:00-23:00	1.87±0.05	2.89±0.05	17.45 ±0.2

information can be treated as useful features and learned by the neural networks. Also, these features can be linked with the velocity change well. Secondly, in the congested situations, due to lots of sudden speed drops accumulated in different road segment, it is difficult to capture the acceleration and deceleration feature effectively through the network. From the result analysis, the overall network prediction of cells with higher speed is better than that at lower speeds. In summary, the PSPNN performs not good enough when there are many acceleration and deceleration happened like peak hour periods. This is also a future topic targeted by the research team.

D. Departure Time Analysis

In the road network, paths with different starting times always lead to different real speed distribution. In order to evaluate and demonstrate the speed prediction accuracy of PSPNN in different time periods, parts of the test data sets (3 days) are divided into different time period according to the starting time. Table II shows the specific results. It can be seen that PSPNN has the ability to predict travel speed around the clock. During the off-peak hours, PSPNN can achieve an accurate travel speed prediction, whose error fluctuation can reach 2-3%. The highest prediction accuracy time domain is during 6:00-7:00. The MAPE is 17.14% and the MAE is 1.83 m/s. However, during the morning peak (7:00-8:00) and the evening peak (18:00-19:00) hours, the prediction error of PSPNN is relatively large. The MAE during the morning peak period is 2.34 m/s, and the MAPE is 22.25%. The MAE is 2.64 m/s in the evening peak period, and the MAPE is 26.02%. The speed prediction errors are relatively large in the morning and evening peak hours. The possible reasons are: 1) serious traffic congestion, 2) due to changes in signal lights, accidents and other external factors, the road network predictability deteriorates, and 3) the acceleration and deceleration are more frequent than other time period whose features are hard to capture.

E. Path Length Analysis

As in Table III, the researchers analyzed the trip to some extent, the accuracy level of PSPNN depends on the lengths of trajectories. For short-distance paths within 5 km, the prediction error of PSPNN is the relatively large (MAPE 23.19% ± 0.15%, MAE 2.17 m/s ± 0.15 m/s). The possible reasons can be summarized as three points: 1) There are few sampling points for paths less than 5 km, and the size of the re-sample is not sufficient for the trajectory characteristics. 2) The speed distribution of shorter journeys is often affected by signal lights, real-time road conditions, so the statistical patterns are not obvious. 3) Some Short path records are always with a higher level haphazardness and uncertainty. For paths length are in 5 km-10 km and 10 km-15 km, that people travel most frequently in the urban area, PSPNN performs excellent prediction results and the error is stable. For the 5-10 km paths, the prediction result (18.11% ± 0.1%, 1.83 m/s ± 0.1 m/s) is the best. When the path length increasing, the prediction accuracy drops slowly since each cell length becomes longer. For the path over 30 km, the size of data in both training and testing sets is so limited that we exclude this part from the experiment.

TABLE III

ANALYSIS OF DIFFERENT PATH DISTANCE AND ERROR

Dis(km)	MAE (m/s)	RMSE (m/s)	MAPE (%)
0-5	2.17±0.15	3.65±0.01	23.19±0.15
5-10	1.83±0.1	2.92±0.05	18.11±0.1
10-15	1.89±0.1	2.96±0.05	18.59 ±0.1
15-20	1.97±0.15	3.06±0.1	21.49 ±0.15
20-25	2.31±0.2	3.35±0.2	26.76 ±0.25
25-30	2.45±0.2	3.83±0.2	29.31 ±0.3
30+	NA	NA	NA

VIII. CONCLUSION AND FUTURE RESEARCH

Individual path-based traffic parameter prediction is a necessary research scope in the transportation field. In this article, researchers present a path-based speed prediction based on deep neural network structure PSPNN for a given path by integrating spatial-temporal and attributes information. With the combination of individual and systematical features, path-based methods can predict flexible and useful information for both travelers and managers. This work narrows down the minimum prediction unit to path cells and achieves an encouraging and promising speed prediction results.

In practice, the path-based speed prediction method proposed in this article can be served for many applications, including e-hailing taxi dispatch optimization, autonomous vehicle route planning, congestion avoidance and etc. Meanwhile, since the PSPNN narrows down the research scope to the individual path level, human behavior and preference can not be neglected. In the real world, human beings are the host of the transportation system. Therefore, to improve the prediction accuracy with human features and interactions can be a future research topic. More useful information will be integrated and extracted for prediction based on the path scope in the future.

ACKNOWLEDGMENT

The authors would like to thank the Didi Chuxing Technology Co, for the open-source dataset – DiDi Chuxing GAIA Open Data Initiative (available from https://gaia.didichuxing.com). The authors would also like to thank the Pacific Northwest Transportation Consortium (PacTrans) at Regional University Transportation Center (UTC) for Federal Region 10, for funding this research.

REFERENCES

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[2] G. F. Newell, "A simplified theory of kinematic waves in highway traffic, part I: General theory," *Transp. Res. Part B, Methodol.*, vol. 27, no. 4, pp. 281–287, Aug. 1993.

[3] M. Ben-Akiva, M. Bierlaire, H. Koutsopoulos, and R. Mishalani, "DynaMIT: A simulation-based system for traffic prediction," in *Proc. DACCORD Short Term Forecasting Workshop*, Delft, The Netherlands, 1998, pp. 1–12.

[4] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. Part C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.

[5] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transp. Res. Part C, Emerg. Technol.*, vol. 13, nos. 5–6, pp. 347–369, Oct. 2005.

[6] Z. Cui, R. Ke, and Y. Wang, "Deep stacked bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," in *Proc. 6th Int. Workshop Urban Comput.*, 2016, pp. 1–9.

[7] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: A path based deep learning approach," *Transp. Res. Part C, Emerg. Technol.*, vol. 100, pp. 372–385, Mar. 2019.

[8] B. Jiang and Y. Fei, "Vehicle speed prediction by two-level data driven models in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1793–1801, Jul. 2017.

[9] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: A probabilistic model fusing multi-source data," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1310–1323, Jul. 2018.

[10] J. Macfarlane, "When apps rule the road: The proliferation of navigation apps is causing traffic chaos. It's time to restore order," *IEEE Spectr.*, vol. 56, no. 10, pp. 22–27, Oct. 2019.

[11] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.

[12] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," *J. de Phys. I*, vol. 2, no. 12, pp. 2221–2229, Dec. 1992.

[13] B. S. Kerner, "The physics of traffic," *Phys. World*, vol. 12, no. 8, p. 25, 1999.

[14] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, "LC-RNN: A deep learning model for traffic speed prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3470–3476.

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

[16] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[17] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*. [Online]. Available: http://arxiv.org/abs/1801.02143

[18] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[19] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intell. Transp. Syst.*, vol. 11, no. 9, pp. 531–536, Nov. 2017.

[20] R. Ke, W. Li, Z. Cui, and Y. Wang, "Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact," *Transp. Res. Rec.*, vol. 267, no. 4, pp. 459–470, 2020.

[21] H. Yang, C. Liu, C. Gottsacker, X. Ban, C. Zhang, and Y. Wang, "Cell-speed prediction neural network (CPNN): A deep learning approach for trip-based speed prediction," in *Proc. 98th Annu. Meeting Transp. Res. Board*, Washington, DC, USA, Jan. 2019.

[22] X. Fu, H. Yang, C. Liu, J. Wang, and Y. Wang, "A hybrid neural network for large-scale expressway network OD prediction based on toll data," *PLoS ONE*, vol. 14, no. 5, May 2019, Art. no. e0217241.

[23] Z. Cui, R. Ke, Z. Pu, X. Ma, and Y. Wang, "Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction," *Transp. Res. Part C, Emerg. Technol.*, vol. 115, Jun. 2020, Art. no. 102620.

[24] Z. Cui, L. Lin, Z. Pu, and Y. Wang, "Graph Markov network for traffic forecasting with missing data," *Transp. Res. Part C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102671.

[25] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 28, 2019, doi: 10.1109/TITS.2019.2950416.

[26] X. Ma, H. Zhong, Y. Li, J. Ma, Z. Cui, and Y. Wang, "Forecasting transportation network speed using deep capsule networks with nested LSTM models," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 16, 2020, doi: 10.1109/TITS.2020.2984813.

[27] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[28] T.-W. Huang, J. Cai, H. Yang, H.-M. Hsu, and J.-N. Hwang, "Multi-view vehicle re-identification using temporal attention model and metadata re-ranking," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, vol. 2, 2019.

[29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[30] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with TrackletNet," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 482–490.

[31] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.

[32] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with conv-LSTM," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.

[33] 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. AAAI*, 2018, pp. 1–8.

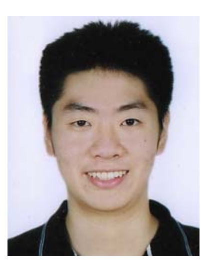
[34] O. Press and L. Wolf, "Using the output embedding to improve language models," 2016, *arXiv:1608.05859*. [Online]. Available: http://arxiv.org/abs/1608.05859

[35] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*. [Online]. Available: http://arxiv.org/abs/1511.07289

[36] S. Lee, Y.-I. Lee, and B. Cho, "Short-term travel speed prediction models in car navigation systems," *J. Adv. Transp.*, vol. 40, no. 2, pp. 122–139, Dec. 2006.

[37] D. Zang, J. Ling, Z. Wei, K. Tang, and J. Cheng, "Long-term traffic speed prediction based on multiscale spatio-temporal feature learning network," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3700–3709, Oct. 2019.

Hao (Frank) Yang (Graduate Student Member, IEEE) received the B.S. degree in telecommunication engineering from the Beijing University of Posts and Telecommunications and the University of London, in 2017. He is currently pursuing the Ph.D. degree with the Department of Civil and Environmental Engineering, University of Washington. He is a Reviewer of Conference on Computer Vision and Pattern Recognition (CVPR), International Conference on Computer Vision (ICCV), Transportation Research Part C: Emerging Technologies, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and *IEEE Intelligent Transportation Systems Magazine*. His research interests include deep learning, computer vision, urban computing, and transportation data science.



Chenxi Liu (Graduate Student Member, IEEE) received the bachelor's degree in civil engineering from Tsinghua University, Beijing, China, in 2017. He came to the University of Washington, Seattle, WA, USA. Since 2017, he has been a Research Assistant with the Smart Transportation Research and Application Laboratory (STARLab). His research interests include studying individual mobility, understanding urban activity pattern by mining big data, data representativeness, and CAV data fusion.

Meixin Zhu received the B.Sc. and M.Sc. degrees in traffic engineering from Tongji University in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree with the University of Washington. He also serves as a Research Assistant with the Smart Transportation Applications and Research Laboratory (STAR Lab), University of Washington, advised by Prof. Yinhai Wang. His research interests include autonomous driving, artificial intelligence, big data analytics, driving behavior, traffic-flow modeling and simulation, and naturalistic driving study.

Xuegang (Jeff) Ban received the B.S. and M.S. degrees in automotive engineering from Tsinghua University and the M.S. degree in computer sciences and the Ph.D. degree in civil engineering (transportation) from the University of Wisconsin at Madison. He is currently a Professor with the Department of Civil and Environmental Engineering, University of Washington. His research interests include transportation network system modeling and simulation, urban traffic modeling and control, and transportation big data analytics, with a focus on emerging technologies and systems in transportation, such as connected/automated vehicle (CAV), new mobility services, and electric vehicles/buses. He is an Editor/Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Transportation Research Part C*, and the *Journal of Intelligent Transportation Systems*, and serves on the editorial board of *Transportation Research Part B*, *Networks and Spatial Economics*, and *Transportmetrica B*. He is a member of the Network Modeling Standing Committee (AEP40) and the Vehicle-Highway Automation Standing Committee (ACP30) of the Transportation Research Board (TRB), under the National Academies of Sciences, Engineering, Medicine. He received the 2011 CAREER Award from the National Science Foundation (NSF), the New Faculty Award from the Council of University Transportation Centers (CUTC), and the American Road and Transportation Builders Association (ARTBA) in 2012.

Yinhai Wang (Senior Member, IEEE) received the master's degree in computer science from the University of Washington (UW) and the Ph.D. degree in transportation engineering from The University of Tokyo in 1998. He is currently a Professor of transportation engineering and the Founding Director of the Smart Transportation Applications and Research Laboratory (STAR Lab), UW. He also serves as the Director of the Pacific Northwest Transportation Consortium (PacTrans), USDOT University Transportation Center for Federal Region 10. He has published more than 190 peer-reviewed journal articles and delivered more than 180 invited talks and nearly 300 other academic presentations. His active research interests include traffic sensing, urban mobility, e-science of transportation, and transportation safety. He serves as a member of the Artificial Intelligence and Advanced Computing Committee of the Transportation Research Board. He is a fellow of the American Society of Civil Engineers (ASCE) and the Past President of the ASCE Transportation and Development Institute (T and DI). He is also a member of the IEEE Smart Cities Technical Activities Committee and was an elected member of the Board of Governors for the IEEE ITS Society from 2010 to 2013. He was winner of the ASCE Journal of Transportation Engineering Best Paper Award in 2003 and the Institute of Transportation Engineers (ITE) Innovation in Education Award in 2018. He is also an Associate Editor of three journals, the *Journal of ITS*, *PLOS One*, and the *Journal of Transportation Engineering*.