Deep Fusion for Travel Time Estimation Based on Road Network Topology

Fuyong Sun, Ruipeng Gao, and Weiwei Xing, *Beijing Jiaotong University, Beijing, 100044, China*
Yaoxue Zhang, *Tsinghua University, Beijing, 100084, China*
Wei Lu, *Beijing Jiaotong University, Beijing, 100044, China*
Jun Fang and Shui Liu, *DiDi Corporation, Beijing, 100089, China*

With the wide application of vehicular location-based services, precise estimation of the travel time plays a crucial role in intelligent transportation systems, such as driving navigation, traffic monitoring, and route planning. Recent methods have made significant progress on public datasets, but are not satisfied for current ride-hailing platforms with complex road network topology and dynamic traffic fluctuation. In this article, we propose an end-to-end Deep Fusion framework for Travel Time Estimation, which exploits multisource heterogeneous traffic information within an encoder–decoder architecture. Specifically, we explore a relational fusion network to learn the relationship of road link segments, and employ an attention mechanism to capture efficient correlations among spatial and temporal features. Extensive experiments have been conducted on two large-scale real-world traffic datasets collected by DiDi Corporation (DiDi) platform, and the results have demonstrated our effectiveness compared with the state of the art.

Travel time estimation is pivotal to many vehicular location-based services such as ride-hailing, vehicle dispatching, and route planning. It not only helps drivers to schedule their trips for better efficiency, but also provides on-demand pick-up services for passengers with better travel experience. An example of travel time estimation on Google map is shown in Figure 1. Given an origin, destination, and departure time, it can predict the riding duration along different paths. Therefore, a series of efforts for travel time estimation have been undertaken in ride-hailing platforms, e.g., Uber, Lyft, and DiDi.

The existing methods on travel time estimation are mainly divided into two categories. One is road segment-based approaches,[1,2] which mainly focus on predicting the travel time on each road segment, then calculate the total time as whole duration time of the origin-destination path. However, these approaches suffer from accumulated errors in prediction, especially in road intersection and traffic lights. Another is the deep learning approaches,[3–6] which exploit neural networks to capture the spatial and temporal correlations in traffic scenarios. Specifically, DeepTTE[4] captures the spatial-temporal relationship based on convolutional neural network (CNN) and long short-term memory (LSTM) via GPS trajectories. Compared with the existing methods, we jointly explore road network topology and road characteristics (e.g., length, width, road class) to capture the spatial correlations. We also involve temporal features to estimate the travel time on road link sequences with a multitask learning mechanism.

Due to the GPS signal drift and the strong spatial correlation of road network topology, in this article, we propose the Deep Fusion framework for Travel Time Estimation (DFTTE), which fully exploits road network topology, road characteristics, and extrinsic contextual information. Such a data fusion approach entails a series of nontrivial challenges, e.g., how to learn the road network in urban cities, and represent the spatial relationship and dynamic traffic of adjacent road segments.

To deal with the above challenges, our contributions include the following.

1541-1672 © 2022 IEEE
Digital Object Identifier 10.1109/MIS.2022.3173427
Date of publication 10 May 2022; date of current version 22 July 2022.

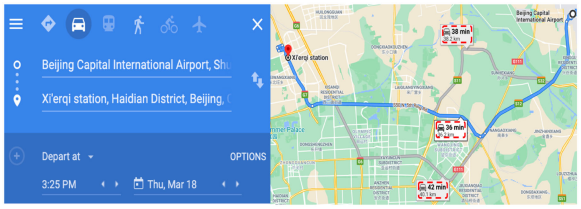

FIGURE 1. Travel time estimation from Google map based on the given origin, destination, and departure time.

1) We propose a relation learning framework on road network to model the hidden topological dependency, which jointly explores the topology and road characteristics to learn the spatial representations of road segments.
2) We design an attention-based encoder–decoder module to fuse spatial and temporal correlations for capturing the spatio-temporal dependency. In addition, we explore an attention-based multitask learning structure to calculate the importance for travel time estimation on both global path and local road segments.
3) We conduct extensive experiments on two large-scale real-world traffic datasets in Beijing and Shanghai, collected by the DiDi ride-hailing platform. Experimental results have demonstrated our effectiveness compared with the state of the art.

RELATED WORK

Travel Time Estimation

The existing methods on travel time estimation are mainly classified into two categories, one is road segment-based methods and another is deep learning methods.

The road segment-based approaches[1,2] predict the transit time during each road segment separately, and then sum them up as the total riding duration of the entire path. Although these methods are effective in some traffic scenarios, they always cause cumulative errors, especially at intersections of adjacent road links.

The deep learning methods[3–6] feed the whole route into prediction model for producing travel time. Specifically, DeepTTE[4] utilized a multitask learning framework for travel time estimation on subpaths and entire path. DeepSTTE[5] leveraged the classical convolution layer and temporal convolutional networks (TCN) to estimate the short-term travel time based on taxis' historical trajectories. Tensor-CNN-LSTM (TCL)[6] proposed a Tensor-CNN-LSTM framework to extract travel speed from historical sparse trajectories and predicted the travel time of a given path. ConSTGAT[7] proposed a spatio-temporal graph attention mechanism to exploit the relations of spatial and temporal information. HetETA[8] combined the gated convolution and graph neural networks to capture the correlations in spatial temporal information.

In practice, the traffic status on each road link is jointly affected by all adjacent road links. Compared with the existing methods, we learn the topology of surrounding road links instead of the ones only along the route. To explore hidden spatial correlations, we effectively model the in/out-link matrix of each road segment, and fuse them with a mask mechanism to learn the correlation of road network topology.

Spatial-Temporal Correlations

In spatial and temporal forecasting, employing deep neural networks and graph neural network to model the spatial-temporal correlations has achieved a significant improvement. Graph Laplacian Regularization method[9] generates the similar representations as adjacent road links in road network topology. Graph convolution networks (GCN)[10] builds the adjacent matrix and feature matrix to model graph data. GWN[11] captured spatial and temporal correlations by combining graph convolution with dilated causal convolution. DC-STGCN[12] proposed a dual-channel based GCNs for network traffic forecasting. GraphSAGE[13] is a general inductive framework that leverages node feature information to generate node embeddings for previously unseen data. Relational Fusion Network (RFN)[14] is a type of GCN for leveraging the road network structure in road segment classification.

Considering the road segments adjacent relationship in traffic scenarios, we proposed the relation leaning, which by building and multiplying road network adjacent matrix, in-links and out-links adjacent matrix, to capture the road segments spatial correlations.

PRELIMINARIES

In this section, we present several important definitions in travel time estimation.

Definition 1 (Road Network): We denote a road network as a directed graph $G = (V, E, A, F_G)$*. Here,* V *is set of nodes (road links) and* E *is the set of edges (connectivity between road links) in the graph.* $A \in \mathbf{R}^{N \times N}$ *is a adjacency matrix, where* N *is the number of nodes and* A_{ij} *is a binary value representing whether two link segments are connected.* $F_G \in \mathbf{R}^{\mathbf{N} \times \mathbf{D}}$ *represents the feature matrix of each road segment, where* D *indicates the feature dimensions of road segments. Given node* v*,* $v \in \mathbf{V}$*, we define* h_v^0 *as its fixed node features, including its length, width, direction, road class (high-speed/city/country), number of lanes, and speed limit.*

Definition 2 (Trajectory and Path): A trajectory T *consists of a sequence of GPS points. Each point*

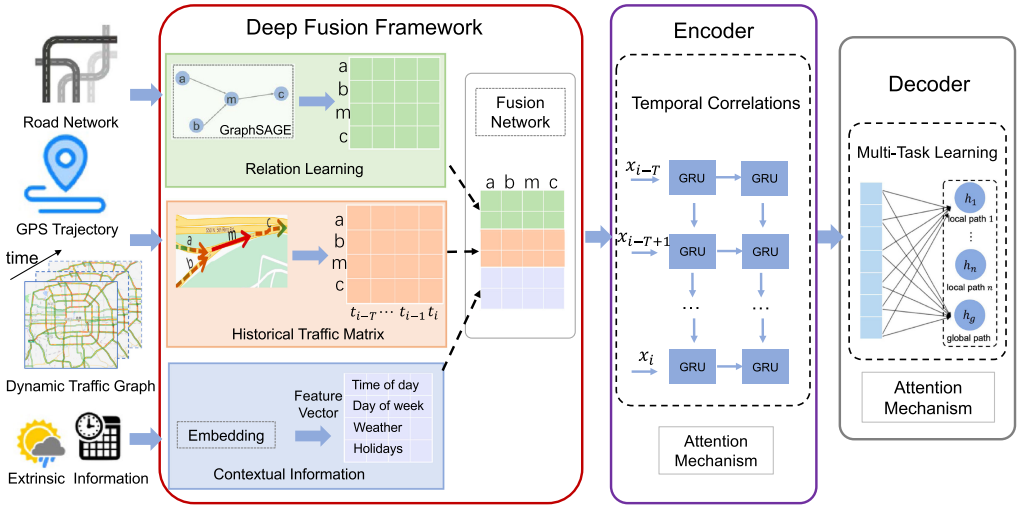

FIGURE 2. Overview of DFTTE. We explore a deep fusion framework and an encoder–decoder structure to learn spatial-temporal correlations for travel time estimation.

contains the timestamp p_t, latitude p_{lat}, longitude p_{lng}, and road link index p_i. The road link index indicates road segment on the map, e.g., $T = (p_t, p_{\text{lat}}, p_{\text{lng}}, p_i)$. A path P is represented by a sequence of connected road links based on trajectory T.

Definition 3 (k-order neighbors): A road link r_i is a k-order neighbors of r_j, if the two road links are connected via k intersections in the shortest path among them. For example, in Figure 5, the direct adjacent links of current-link are in-link and out-link, which are called 1-order neighbors of current-link.

Travel Time Estimation: Given a query $q = (o_q, d_q, t_q)$, where o_q is the origin of query, d_q is the destination of query, t_q is the departure time, and our aim is to estimate the travel time based on historical trajectory datasets and underlying road network.

PROPOSED APPROACH

The overall architecture of DFTTE is presented in Figure 2. It mainly consists of deep fusion framework and encoder–decoder structure. Specifically, we jointly explore the road network topology and the road characteristics (e.g., length, width, road class) to capture the spatial correlations based on relation learning method. Then, we combine the temporal features and contextual information to estimate the travel time on road link sequences with multitask learning mechanism.

Deep Fusion Framework

The deep fusion framework mainly focuses on capturing the spatial dependency with considering the underlying road network topology. It mainly includes the extrinsic contextual information and relation learning components. The details are as follows.

Extrinsic Contextual Information

Travel time is affected by extrinsic contextual information in daily transportation, such as weekdays, weekends, and weather conditions.

First, Figure 3 shows the comparison of travel time during one week. We observe that there is a common tendency and periodicity of general working hours for most residents in traffic scenarios. As shown in Figure 3, there is a consistent rise of travel time at 7–9 a.m. and 5–7 p.m. during weekdays, but it remains almost stable at weekends.

Second, in Figure 4, we visualize the travel time with different weather conditions, one is cloudy and another one is shower. It depicts the effects of two frequent weather conditions in summer for travel time estimation. The red line represents the average travel time

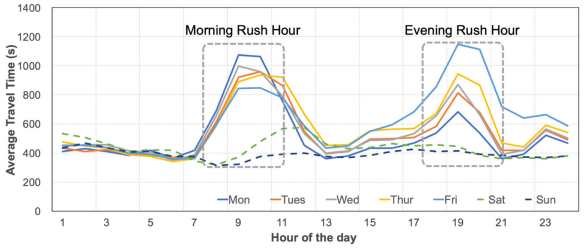

FIGURE 3. Travel time fluctuation of weekdays and weekends.

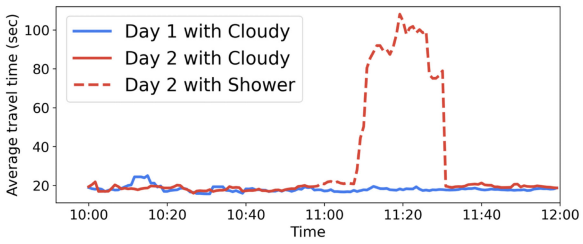

FIGURE 4. Comparison for different weather conditions.

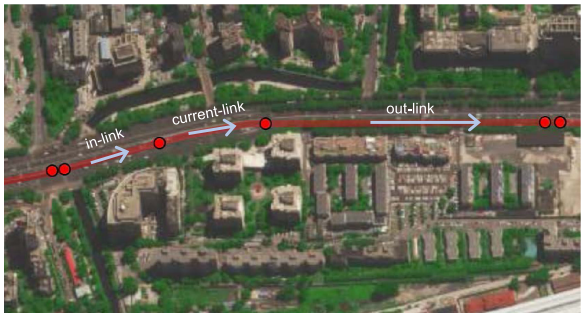

FIGURE 5. Adjacent road links in one route.

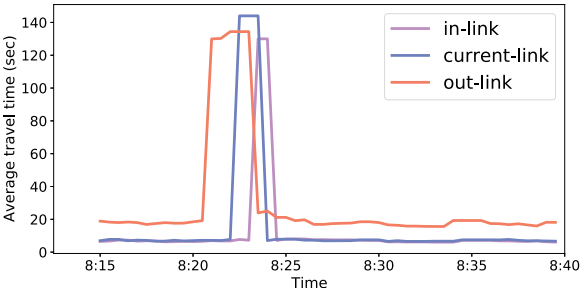

FIGURE 6. Fluctuation of traffic congestion among adjacent road links.

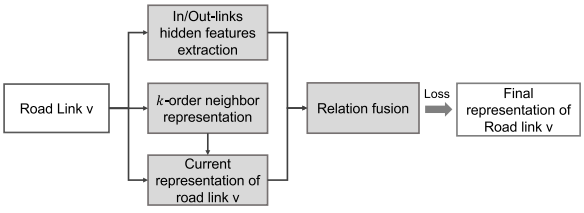

FIGURE 7. Flowchart of the relation learning algorithm.

fluctuation of weather from cloudy to shower, then to cloudy in one day. The red dotted line indicates the shower. We observe that the shower increases travel time with more than 60 s compared to cloudy.

However, since these contextual information is a binary value, it cannot be directly fed into the deep learning model. Therefore, we employ an embedding method[4] to transform the contextual features into low-dimensional feature vectors as V_{cf}. In our experiment, we embed the day of the week to R^7, weather condition to R^6, and holidays to R^2.

Relation Learning

Travel time is also strongly affected by underlying road network topology, especially among adjacent road links. For example, as shown in Figure 5, the satellite image of three adjacent road links within one route in Beijing. We denoted as the in-link, out-link and current link, respectively.

To verify the effect of adjacent road links, we have collected average travel time of each road link, and then visualized the travel time fluctuation in Figure 6. Specifically, the *x*-axis is time in one day, the *y*-axis is the range of average travel time. When a congestion occurred in out-link, it would also affect its upstream sections, such as current-link and in-link. Therefore, we propose a relational fusion method to capture spatial correlations of each road link.

TABLE 1. Experimental parameters.

Parameters	Description
h_v^0	The node features of node v
$h_{\text{in/out}}^k$	The hidden feature vectors of the k-order in/out-links neighbors
$M_{\text{in/out}}^k$	The k-order in/out-link adjacent matrices
$F_{\text{in/out}}$	The in/out-link feature matrices
W^*	The learnable weight matrices
z_v	The final representation of road link v

Specifically, given road network graph $G = (V, E)$, we utilize K aggregator functions (denoted as AGGREGATE_k, $\forall k \in \{1, \ldots, K\}$) to aggregate information from adjacent nodes, and the learning weight matrices W^k, $\forall k \in \{1, \ldots, K\}$ to propagate information from different layers. Note that the k also indicates k-order neighbor road links and the representations of $k = 0$ are defined as the input node features. We present the flowchart of the relation learning algorithm in Figure 7. The gray blocks indicates the four steps to learn the spatial correlations (from i) to iv)).

We first present the experimental parameters in Table 1 to improve the clarity.

Then the details are as follows.

i) We model the relations of each road segment based on road network topology. Specifically, we build

the k-order in/out-link matrices $M^k_{\text{in/out}}$ for each road segment according to its adjacent road links, and then combining the feature matrices $F_{\text{in/out}}$ to obtain the formalization k-order hidden features of in/out-links $h^k_{\text{in/out}}$ by fully connected layers (FCLs) as follows:

$$h^k_{\text{in}} = \sigma(M^k_{\text{in}} \cdot F_{\text{in}}) \tag{1}$$

$$h^k_{\text{out}} = \sigma(M^k_{\text{out}} \cdot F_{\text{out}}) \tag{2}$$

where σ is activation function, and F_{in} and F_{out} represent the feature matrix of in-links and out-links, i.e., length, width, direction.

ii) We extract the k-order neighbor road links of current road segment by kth layer to attain the representation of neighbors. Each road link v aggregate the representations of road links in its k-order neighborhood $N(v)$ to the neighborhood vector $h^k_{N(v)}$, which is generated in previous iteration:

$$h^k_{N(v)} = \text{AGGREGATE}_k(\{h^{k-1}_u \quad \forall u \in N(v)\}) \tag{3}$$

where h^{k-1}_u means the hidden feature vector of $(k-1)$-order neighbors with $k-1$ layers for road link u in $N(v)$.

iii) We concatenate the current road link's representation h^{k-1}_v and its aggregated neighborhood vector $h^k_{N(v)}$, then feed it into FCLs with a nonlinear activation function σ for the representations of the next step such that

$$h^k_v = \sigma(W^k \cdot \text{CONCAT}(h^{k-1}_v, h^k_{N(v)})), k \geq 1 \tag{4}$$

where $\text{CONCAT}(\cdot)$ is a concatenate function with specified dimension, and σ is a nonlinear activation function.

iv) We design a relational fusion method to capture spatial correlations. Specifically, we take the in/out-links hidden feature vectors and road links vector into (5) for final representation of road link v, which is calculated as z_v (i.e. V_{sf}):

$$z_v = \sigma(W^v h^k_v + W^{\text{in}} h^k_{\text{in}} + W^{\text{out}} h^k_{\text{out}} + b) \tag{5}$$

where σ is a nonlinear activation function, and h^k_v indicates the representation of k-order's road link v. W^v is the current road links weight matrix, $W^{\text{in/out}}$ is in/out-links weight matrices, and b is a bias term.

In order to learn effective representations, the relation learning applies a graph-based loss function $J(z_v)$ to ensures the similarity of adjacent road links and distinction among disparate road links. Compared with previous embedding approaches, the road link representation z_v in this loss function is generated from the features contained within the neighbor road links, rather than an independent embedding for each road link.

$$J(z_v) = -\log(\sigma(z_v^\top z_u)) - Q \cdot E_{u_n \sim P_n(u)} \log(\sigma(-z_v^\top z_u)) \tag{6}$$

where u is a road link that co-occurs near v on k-order neighborhoods, σ is the sigmoid function, P_n is a negative sampling distribution of neighbor road link set of u_n, and Q is the number of negative samples.

Encoder–Decoder Structure

In this section, we integrate multisource features to estimate the travel time. Specifically, V_{sf} is spatial correlation features based on relation learning, the temporal correlation features V_{tf} are comprised of historical traffic speed and travel time on each road segments at different time slots of day. Specially, we divide one day into 720 timeslots with 2-min time step. In addition, we encode the day of week, holidays, weather information as the extrinsic contextual features V_{cf}. Then, we fuse them as V_{all} to learn the spatial temporal dependency

$$V_{\text{all}} = V_{\text{sf}} \parallel V_{tf} \parallel V_{\text{cf}} \tag{7}$$

where $\parallel$ is a vector concatenation operation.

Moreover, we explore the attention mechanism and multitask learning to improve the predictive accuracy. The details are as follows.

Encoder Component

In application of learning time series, it has been demonstrated that a stacked GRU[a] is effective to boost the generalization ability.[15]

To learn the temporal correlations among different local paths, we utilize two layers of GRU as the encoder. It aims to learn the hidden long- and short-term temporal correlations based on historical traffic data. Given a fused feature sequence $X = (x_1, x_2, \ldots, x_T)$, the GRU learns a mapping from x_t to h_t at time step t. Specially, $x_t = V^t_{\text{all}}$, represented as an input feature at time step t, i.e.,

$$h_t = \text{GRU}(h_{t-1}, x_t) \tag{8}$$

where h_{t-1} is a hidden state of the GRU at time step $t-1$.

Attention Mechanism

To model the dynamic correlations of spatial and temporal features, we utilize the attention mechanism[4] to learn weights in different traffic conditions.

The spatial and temporal attention mechanism is the weighted sum of fused feature V_{all}, i.e., the final

[a][Online]. Available: https://www.worldcat.org/title/deep-learning/oclc/955778308

features h_{all} is calculated as

$$h_{\text{all}} = \sum_j a_j \cdot h_j \tag{9}$$

where a_j is an attention weight for jth hidden feature of local path, h_j is the hidden feature of local path in GRU, and the sum of attention weight is $\sum_j a_j = 1$.

In addition, the attention weight a_j is calculated as

$$z_j = \text{ReLU}(W_j \cdot h_j + b_j) \tag{10}$$

$$a_j = \frac{\exp(z_j)}{\sum_j \exp(z_j)} \tag{11}$$

where W_j is a weight matrix, and b_j is a bias item in attention mechanism.

Decoder Component

We predict travel time with a multitask learning structure. Its input includes the time relevant features on local paths and the global path. We feed it into FCL:

$$o = \text{ReLU}(W_i^{(o)} h_{\text{all}}) \tag{12}$$

where $W_i^{(o)}$ is the parameter matrix in ith FCL layer.

Multitask Learning Component: In the training phase of our DFTTE model, we exploit the attention-based multitask learning mechanism. We calculate the different weights for local road segments to learn the spatio-temporal correlations. It not only estimates the travel time of each local path, but also predicts the overall time along the global path. In the test phase, DFTTE skips local paths and predicts the travel time for the global path directly.

Loss Function: During the training phase, we employ mean absolute percentage error (MAPE) as our objective function for global path and local paths in travel time estimation. Given a training dataset D, the loss function of the global path L_g is calculated as

$$L_g = \frac{1}{N} \sum_{i \in D_N} \frac{|\hat{y}_{D_i} - y_{D_i}|}{y_{D_i}} \tag{13}$$

where D_i indicates the ith training sample, N is the number of training samples, y_{D_i} is the ground truth for global travel time, and $\hat{y}_{D_i}$ denotes the prediction.

For local paths, the loss function L_l is calculated as

$$L_l = \frac{1}{N} \sum_{i \in D_N} \sum_{j \in n} \frac{|\hat{y}_{D_i^j} - y_{D_i^j}|}{y_{D_i^j}} \tag{14}$$

where n denotes the total number of road links in each D_i training sample, $\hat{y}_{D_i^j}$ denotes the estimate travel time, and $y_{D_i^j}$ is the ground truth of jth link segment in local path. We set the loss function of our model as L in the following equation, which combines the global path L_g and local path L_l with a tradeoff combination coefficient β.

$$L = \beta L_g + (1 - \beta) L_l. \tag{15}$$

TABLE 2. Statistics of datasets.

Dataset	Beijing	Shanghai
Time range	68.6-8.26 2018	8.6-8.26 2018
No. of trajectories	235,172	109,035
No. of nodes ($\|V\|$)	8963	6532
No. of edges ($\|E\|$)	12537	9716

EXPERIMENTS

In this section, we evaluate the performance of DFTTE on two large-scale real traffic datasets from DiDi ride-hailing platform. Specifically, we first ensure the effectiveness of relation learning in (6), and then, we adopt (15) to optimize our travel time estimation model with both local road segments and global path in training phase.

Experimental Data Description

Our experimental datasets are gathered in Beijing and Shanghai. The time period is from Aug. 6 to Aug. 26, 2018.

The dataset mainly contains GPS trajectories, road network, and contextual information (like weather condition, holidays, etc.). Owing to GPS signal drift, we first utilize the MapMatching algorithm[16] to map GPS points into road network, and then extract road link sequences of path from trajectories. Meanwhile, the data range of different road network attribute features is not specific, so we utilize Z-score to normalize it in data preprocessing. Table 2 shows the statistics of the experimental datasets.

Experimental Settings

In experiment, we utilize 60% of the data for training, 20% for validation, and 20% for testing. The time step of traffic datasets denoted as 2 min.[b] The hidden size of two stacked GRU is 128, the tradeoff combination coefficient β in multitask learning component is 0.4. In addition, various number of adjacent neighbor road links have different impact on traffic fluctuation. To alleviate this side effect, we utilize the L_2 normalization to scale it, and build the 3-order matrix with mask mechanism. During the training phase, we utilize the Adam optimizer[c] with a learning rate of 0.001. We conduct experiments based on PyTorch deep learning

[b]We divide one day into 720 timeslots with 2-min time step.
[c][Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

TABLE 3. Mean and standard deviations of different approaches for travel time estimation in Beijing and Shanghai.

Methods	Beijing			Shanghai		
	MAPE(%)	MAE (s)	RMSE (s)	MAPE (%)	MAE (s)	RMSE (s)
SVR	30.25 ± 0.00	282.6 ± 0.00	532.8 ± 0.00	36.15 ± 0.00	377.5 ± 0.00	605.7 ± 0.00
XGBoost	28.12 ± 0.00	266.6 ± 0.00	464.3 ± 0.00	31.43 ± 0.00	348.6 ± 0.00	556.2 ± 0.00
LSTM	27.92 ± 0.21	264.3 ± 1.57	457.6 ± 1.83	29.25 ± 0.15	321.3 ± 1.39	512.3 ± 2.17
DeepTTE	21.38 ± 0.09	227.5 ± 1.39	426.4 ± 2.15	25.21 ± 0.19	297.5 ± 1.85	468.5 ± 2.57
DeepETA	23.53 ± 0.17	232.4 ± 1.15	441.8 ± 1.68	27.27 ± 0.21	306.2 ± 1.01	493.7 ± 2.16
DeepTravel	22.15 ± 0.12	228.3 ± 1.28	435.7 ± 2.11	25.58 ± 0.15	299.5 ± 2.03	476.3 ± 2.25
GWN	21.27 ± 0.13	235.2 ± 0.92	412.5 ± 1.29	23.62 ± 0.18	291.7 ± 1.39	457.6 ± 2.01
GWN-F	19.75 ± 0.08	210.8 ± 0.83	384.4 ± 1.58	21.35 ± 0.10	268.2 ± 1.12	406.2 ± 1.73
Ours	**17.32 ± 0.11**	**207.5 ± 0.65**	**346.5 ± 1.49**	**20.26 ± 0.08**	**255.8 ± 1.31**	**372.8 ± 1.58**

framework, and train the DFTTE model on a 64-bit server with NVIDIA GTX 2080Ti.

We evaluate the predictive performance on our proposed model with three criterions, *mean absolute percentage error (MAPE)*, *mean absolute error (MAE)*, and *rooted mean square error (RMSE)*.

Methods for Comparison

To demonstrate the effectiveness of our method, we compare Ours (i.e., DFTTE) with the following baseline methods. In addition, we preserve the default experimental settings and model framework in original papers.

1) Support vector regression (SVR)[17]: It is widely used in sequence prediction based on SVM.
2) XGBoost[18]: It is a scalable tree boosting system and widely used on time series prediction.
3) LSTM[19]: It has been demonstrated the effectiveness to capture time series information. We utilize LSTM with two stacked layers and set the hidden size as 128.
4) DeepTTE[4]: It utilized a geo-convolution, two layers of LSTM and attention mechanism to capture spatial and temporal dependencies for travel time estimation.
5) DeepETA[20]: It proposed a wide and deep neural network with a spatial-temporal module to capture the time series features for travel time estimation.
6) DeepTravel[3]: It simultaneously extracted spatial and temporal features and then employs dual interval loss to leverage the temporal information based on bidirectional LSTM.
7) GWN[11]: It captured the spatial and temporal correlations by combining GCN and gated temporal convolution module (Gated TCN). Since our fusion matrix integrates the *k*-order in/out-links of each road segment and its attributes for spatial learning, we use it to replace the self-adaptive adjacency matrix in GWN and denote it as GWN-F.

Experimental Results

In this section, we conduct extensive experiments on two large-scale traffic datasets in Beijing and Shanghai. The results of different approaches are shown in Table 3. We observe that the MAPE value of GWN-F is nearly 2% lower than original GWN. The DFTTE is nearly 3.5% lower than GWN on MAPE. Overall, the DFTTE achieves a better performance in both Beijing and Shanghai datasets.

In addition, we visualize the model performance in Beijing dataset from Figures 8 to 12.

Effect of Extracted Features

The ablation study on GPS trajectories and road network is shown in Figure 8. We can observe that only utilizing the road network has the high value of MAPE, that is because some road segments have sparse GPS trajectories. As a result, our method achieves a lower value with considering the spatial and temporal correlations.

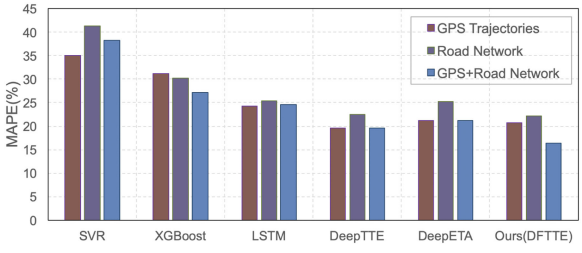

FIGURE 8. Ablation study on GPS trajectories and road network.

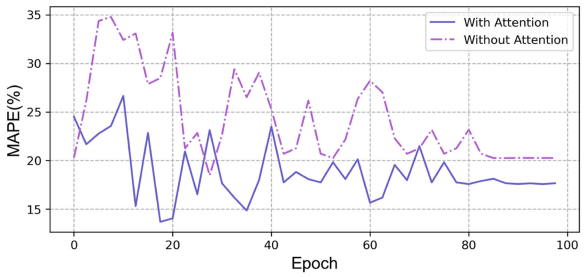

FIGURE 9. Effect of attention mechanism on Training MAPE.

Effect of Attention Mechanism

Attention mechanism calculates different weights to distinguish the most relevant features. The comparison results are shown in Figure 9, we can observe that the value of training MAPE with attention mechanism is obviously lower than without attention during the training phase, and the model with attention mechanism converges faster. With the epoch increasing, the value first fluctuates in a small range, then gradually converges to a small value, and finally achieves a stable state. As a result, the attention mechanism is helpful for improving the prediction performance.

Model Interpretation

In order to investigate the predictive ability of DFTTE, especially on weekdays and weekends, we set the time window size as 10 minutes and one hour. We compare the predictions of the DFTTE with the ground truth values of one route in the test traffic data via visualization from Figures 10 and 11 separately. We analyze the predictive performance from two aspects, one is long short-term prediction, and another is extrinsic contextual information.

a) Long short-term prediction: We forecast and visualize travel time at different time intervals to verify the robustness of proposed model. In Figures 10(a) and 11(a), we can see that the model performance on different time interval. The abscissa is the date in one week, and the ordinate is the average travel time. The blue line indicates the ground truth of travel time, and the purple line indicates the predicted result. The shaded box represents the travel time estimation on weekday (Thursday) and weekend (Sunday) at different time intervals, separately. First, for the long-term prediction, we set the time interval as one hour. The corresponding details are shown in Figure 10(b) and (c). And we can observe that the model is robust in capturing the long-term trend of traffic

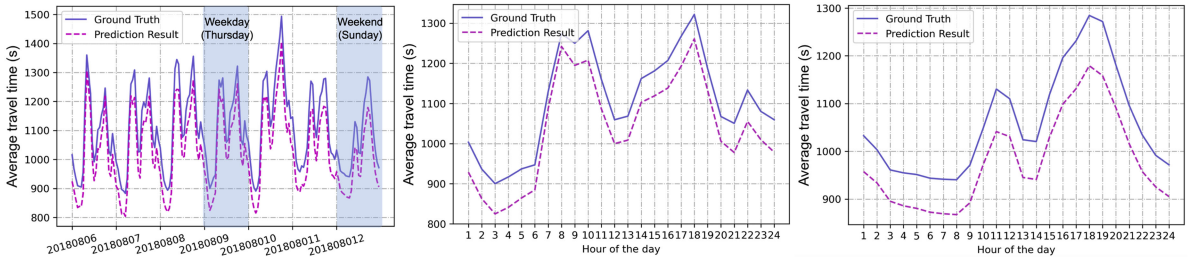

FIGURE 10. Travel time estimation at an one-hour time interval. (a) Travel time estimation during one week. (b) Travel time estimation on weekday (Thursday). (c) Travel time estimation on weekend (Sunday).

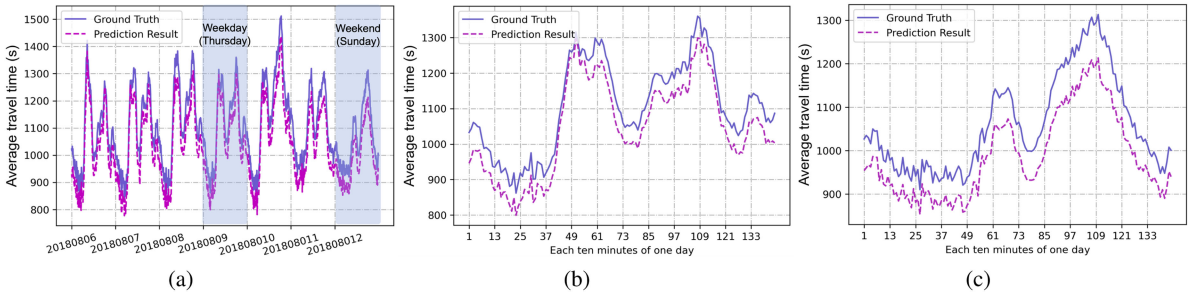

FIGURE 11. Travel time estimation at a ten-minute time interval. (a) Travel time estimation during one week. (b) Travel time estimation on weekday (Thursday). (c) Travel time estimation on weekend (Sunday).

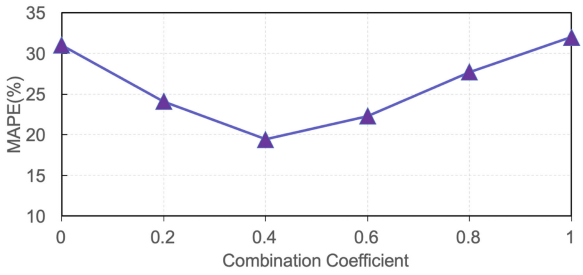

FIGURE 12. Effect of weight combination coefficient.

fluctuation. Second, we set the time interval as ten minutes for the short-term prediction. The result is shown in Figure 11. The details of ten minutes prediction are shown in Figure 11(b) and (c). From the visualization, we know that the proposed model has the ability of capturing the traffic fluctuation in a short-time interval.

b) Extrinsic contextual information: The transportation behavior is various under different contextual information, such as the different traffic patterns on weekdays or weekends. Therefore, we analyzed and conducted the experiments on weekdays and weekends, respectively. The performances are shown in Figures 10 and 11. We observe that there is a common tendency in urban traffic scenarios that the travel time predictions on weekdays are better than weekends. We think this is due to the periodicity of general working hours for most residents. As shown in Figure 3, there is a consistent rise of travel time at 7–9 a.m. and 5–7 p.m. during weekdays, but it remains almost stable at weekends. Our model effectively learns spatio-temporal features to capture the periodic fluctuation of traffic; therefore, it produces more precise predictions on weekdays.

c) Hyperparameters: In a multitask learning component, we evaluate our model under different tradeoff combination coefficient β from 0.0 to 1.0 between L_g and L_l. The result is shown in Figure 12, When β is 0.4, the model achieves a lower MAPE with jointly considering the interaction of local path and global path. When β is 0, the MAPE value of model is larger than 30%, which lacks considering the global path relationship and has a cumulative error. And when we set the β as 1, the model estimates travel time without considering the correlations of local path, so under the situations, the model has a high value of MAPE.

CONCLUSION

In this article, we propose DFTTE to learn road network topology for travel time estimation. It efficiently captures the spatial and temporal correlations in heterogeneous traffic data. Moreover, we design an attention-based multitask learning structure to calculate the weights for learning the travel time on global and local paths in the encoder–decoder architecture. We have conducted extensive experiments on two large-scale real-world traffic datasets in Beijing and Shanghai, and the results demonstrate the effectiveness of proposed method. However, in the future work, there are still many fields to improve, e.g., enhancing the stability at the initial training stage, and investigating the effect of traffic incidents for travel time estimation.

ACKNOWLEDGMENTS

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2021YJS185, in part by Beijing National Science Foundation under Grant L192004, in part by the National Natural Science Foundation of China under Grants 61876017 and 62072029, and in part by the DiDi Research Collaboration Plan.

REFERENCES

1. Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov Data Mining*, 2014, pp. 25–34.
2. 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, 2019.
3. H. Zhang, H. Wu, W. Sun, and B. Zheng, "Deeptravel: A neural network based travel time estimation model with auxiliary supervision," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3655–3661.
4. 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 Conf. Artif. Intell.*, 2018, vol. 18, pp. 1–8.
5. L. Fu, J. Li, Z. Lv, Y. Li, and Q. Lin, "Estimation of short-term online taxi travel time based on neural network," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.*, 2020, pp. 20–29.
6. Y. Shen, J. Hua, C. Jin, and D. Huang, "TCL: Tensor-CNN-LSTM for travel time prediction with sparse trajectory data," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2019, pp. 329–333.
7. X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov Data Mining*, 2020, pp. 2697–2705.

8. H. Hong *et al.*, "Heteta: Heterogeneous information network embedding for estimating time of arrival," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov Data Mining*, 2020, pp. 2444–2454.
9. Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov Data Mining*, 2018, pp. 1695–1704.
10. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907v4*.
11. Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.
12. C. Pan, J. Zhu, Z. Kong, H. Shi, and W. Yang, "DC-STGCN: Dual-channel based graph convolutional networks for network traffic forecasting," *Electronics*, vol. 10, no. 9, 2021, Art. no. 1014.
13. W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
14. T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Relational fusion networks: Graph convolutional networks for road networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 418–429, Jan. 2022.
15. S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "DeepSense: A unified deep learning framework for time-series mobile sensing data processing," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 351–360.
16. Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang, "Map-matching for low-sampling-rate GPS trajectories," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2009, pp. 352–361.
17. C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.
18. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov Data Mining*, 2016, pp. 785–794.
19. 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, 2015.
20. F. Wu and L. Wu, "DeepETA: A spatial-temporal sequential neural network model for estimating time of arrival in package delivery system," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 774–781.

FUYONG SUN is currently working toward the Ph.D. degree with the School of Software Engineering, Beijing Jiaotong University, Beijing, China. His research interests include deep learning and intelligent transportation system. Contact him at fysun12@bjtu.edu.cn.

RUIPENG GAO is an associate professor with the School of Software Engineering, Beijing Jiaotong University, Beijing, China. His research interests include mobile computing and applications, and intelligent transportation systems. Gao received his Ph.D. degree from the Peking University, China. Contact him at rpgao@bjtu.edu.cn.

WEIWEI XING is a professor with the School of Software Engineering, Beijing Jiaotong University, Beijing, China. Her research interests include software engineering, intelligent information processing, and intelligent transportation. Xing received her Ph.D. degree in computer science from Beijing Jiaotong University. She is the corresponding author of this article. Contact her at wwxing@bjtu.edu.cn.

YAOXUE ZHANG is a professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include computer networks, operating systems, ubiquitous/pervasive computing, and Big Data. Zhang received his Ph.D. degree in computer networking from Tohoku University, Japan. He is a fellow of the Chinese Academy of Engineering. He is the co-corresponding author of this article. Contact him at zhangyx@tsinghua.edu.cn.

WEI LU is a professor with the School of Software Engineering, Beijing Jiaotong University, Beijing, China. His research interests include computer networks and multimedia information processing. Lu received his Ph.D. degree in information and communication engineering from Sichuan University, Chengdu, China. Contact him at luwei@bjtu.edu.cn.

JUN FANG is the chief algorithm engineer with the Maps and Public Transportation Department, DiDi Corporation, Beijing, China. His research interests include machine learning, deep learning, and spatio-temporal forecasting. Fang received his M.S. degree in computer science and technology from the University of Science and Technology of China, Hefei, China. Contact him at fangjun@didiglobal.com.

SHUI LIU is the chief algorithm engineer with the Maps and Public Transportation Department, DiDi Corporation, Beijing, China. His research interests include natural language processing, deep learning, and spatio-temporal forecasting. Liu received his Ph.D. degree from the Harbin Institute of Technology, China. Contact him at liushui@didiglobal.com.