Applied Intelligence (2023) 53:30384–30399
https://doi.org/10.1007/s10489-023-05139-6

TransETA: transformer networks for estimated time of arrival with local congestion representation

Shu Lin[1] · Yanyan Xu[2] · Shengjian Zhao[1] · Yibing Wang[3] · Jungang Xu[1]

Accepted: 27 October 2023 / Published online: 17 November 2023

Abstract

Estimated time of arrival (ETA) is an estimate of the vehicle travel time from the origin to destination in the roadworks. From the perspective of travel planning or resource allocation, accurate ETA is significantly important. In recent years, deep learning-based methods represented by recurrent neural networks has been widely used in travel time prediction tasks, but such methods cannot effectively learn data association at different moments. At the same time, the existing methods do not effectively leverage local traffic information. Targeting these challenges, this paper proposes a new model TransETA to predict vehicle travel time. The model includes three modules: the input feature transformation module uses graph convolutional network (GCN) to extract the local congestion feature, the deep forest module mainly deals with static trajectory data, and ETA-Transformer module processes the feature extraction of dynamic trajectory data. Finally, we conducted experiments on two large trajectory datasets. The experimental results show that the proposed hybrid deep learning method, TransETA, outperforms the state-of-the-art models. On the Chengdu and Porto datasets, our proposed method shows an improvement of **6s** and **9s** in mean absolute error compared to the current best performing method, respectively. Also the average absolute percentage error is reduced by **2.34%** and **3.64%** respectively. The effectiveness of each module was approved through ablation experiments. Specifically, local congestion information representation can effectively improve the accuracy of the prediction. ETA-Transformer module is more effective in extracting spatio-temporal feature correlation than the LSTM-based method.

Keywords Travel time prediction · Transformer · Traffic flow · Graph convolutional network

1 Introduction

Travel time prediction refers to the vehicle travel time estimation between origin and destination locations. Estimated time of arrival (ETA) is one of the most important location-based digital map and navigation system services. It is widely used in taxi-hailing platforms, takeaway delivery, and public transportation. At the same time, with the popularization of various mobile devices equipped with GPS sensors, massive amounts of vehicle trajectory data have been generated and collected. These data contain important information about urban travel, making it possible to build better intelligent transportation systems (ITS) to reduce traffic congestion and improve people's daily commuting efficiency.

In recent years, deep learning has made breakthroughs in computer vision, natural language processing, and other fields, indicating that deep learning has a strong representation ability for various types of data. For traffic trajectory data, the trajectory points are sequentially collected at a certain time interval, which is similar to sentence sequence features in natural language processing. The road network has a natural graph structure, which is very suitable for graph convolutional network processing. Therefore, accurate prediction of travel time based on deep learning requires the integration of technologies in various fields, which is a challenging task [1]. In recent years, the prediction of travel time has also shifted from traditional statistical methods to machine learning and deep learning, especially in the field of deep learning. Due to the powerful learning ability of deep learning for massive data and different network structures have their different applicable scenarios, a number of

✉ Shu Lin
s_lin_wy@163.com

✉ Yanyan Xu
yanyanxu@sjtu.edu.cn

Extended author information available on the last page of the article

integrated neural networks have become the research focus to achieve accurate prediction effect [2]. For example, the design of the attention mechanism can effectively capture the front and back timing correlation of traffic trajectory data; the modeling of the graph convolutional network (GCN) on road networks can learn the variations of road flow dynamics etc. Deep learning is a fast developing state-of-the-art methodology, which can integrate temporal and spatial information all together. Therefore, this method is very suitable to be applied to predict the travel time for traffic networks. Travel time prediction problems can be divided into three categories according to the modeling methods that used, which has experienced an era from statistical learning to machine learning and finally to deep learning. Travel time prediction method based on statistics learning predicts the future travel time with time series prediction methods based on the historical traffic data. However, this kind of method can only predicts the travel time of fixed origins and destinations, since it does not take into account the spatial information. By applying decomposition methods, the time series data can be divided into multiple sub-sequences, including long-term trends, seasonal changes, and cyclical changes. Normally, an additive model or a multiplicative model, which are applied to the decomposed sub-sequences, are combined to obtain the final prediction results. But, this type of methods is mainly appliedd for the modeling of non-stationary series. After that, the prediction model named as autoregressive integrated moving average model (ARIMA) [3] appeared, which is a combination of autoregressive model (AR) [4], moving average model (MA) [5] and difference methods [6]. Compared with single models, ARIMA can process both stationary series and non-stationary series. Both of the above mentioned two methods require a lot of artificial parameter adjusting, and noise data normally have a big impact on the prediction results. Then, machine learning-based methods, such as support vector machines [7] and decision trees [8], began to appear. They can learn both linear and non-linear characteristics of a system, but this type of method requires a lot of efforts in feature extraction and processing before making predictions. Useful features are selected to input into the model through feature analysis, feature combination, and other operations. Experiments need to spend a lot of time on feature analysis and processing [9]. It means that feature analysis and processing is the key factor that influences the prediction results of the machine-learning-based methods. Therefore, XGBoost [10] was proposed as a tree-based ensemble learning method, which applies multiple decision trees to determine the final result jointly. By doing so, the accuracy and robustness of the prediction models can be further improved.

With the development of neural networks, end-to-end models based on deep learning became the main method for ETA problems, which not only improved the accuracy but also increased the iteration efficiency of the model.The prediction methods based on deep learning, e.g. multilayer perceptron (MLP) [11], recurrent neural networks (RNN) [12], and long short-term memory (LSTM) [13], open the door to using deep learning to predict travel time. As a state-of-the-art method in the current ETA task, the LSTM-based method uses the gating unit to avoid disappearing or exploding problems in the process of gradient propagation compared with RNN [1].

From the perspective of how the results are calculated, the existing deep-learning-based prediction methods can be divided into two categories. The first category is path-based solutions [14, 15], which use a physical model to describe the road travel time, and the total travel time of a given route is expressed as the sum of each road segment and the delay time at all intersections. According to the different temporal and spatial data characteristics of road segments and intersections, models are established respectively, such as dynamic Bayesian network and LSTM, where the results of each part are added to obtain the final travel time of a route. This method has strong interpretability, but due to the error accumulation effect of all the predictions of different road segments, the accuracy of the path-based solutions is not high and it cannot be applied to predict the travel time with unknown route information. The second category is a data-driven method that uses location-based trajectory data to build rich features, and divides the data into static data and dynamic data to directly predict the travel time from the start point to the destination, based on end-to-end forecasting methods. This type of method is currently the most accurate and popular prediction method, however, it also has two main problems. First, the traffic flow information is not fully utilized. Some methods do not consider the trajectory information. Some methods consider the trajectory location points, but only map the trajectory location points to one area, rather than a specific road section, which causes the inaccuracy in the traffic flow estimation. Second, this method mainly uses the LSTM-based method in the time dimension, which has information loss during the long-time-distance propagation back into the past, and the processing speed is comparatively slow. In addition, [16] proposes a sustainable transportation planning scheme based on traffic congestion, integrated social, economic, environmental and other factors through network data envelopment analysis (DEA). At the same time, it considers traffic congestion under stochastic and fuzzy conditions to solve the uncertainty conditions in traffic congestion, and establishes a special function to analyze traffic congestion.

In response to the above-mentioned challenges, this paper proposes a model named TransETA to predict vehicle travel time, in which GCN is used in the input feature transformation module to extract road flow features, which are the most important features that influence the vehicle travel time in traffic networks. First, the trajectory data is mapped to the road network through the map-matching method. Then, the number of statistical trajectory points is used as a proxy for the current road flow state. We introduce the proxy to solve the problem that the previous method could not accurately input the road flow characteristics; ETATransformer module uses the transformer-based model to perform feature extraction on dynamic trajectory data to solve the problem of information loss in the hidden layer training process, which can effectively learn the spatial and temporal relationship of data in the sequence through strategies such as multi-head attention. In addition, the feature extractions are handled separately for static data and driving data for a better performance. Deep forest module is selected to learn the features of static traffic data in urban traffic networks, due to its advantage of dealing data from different domains and robustness for hyper-parameter settings. Through the joint improvement of data feature extraction and model structure, TransETA outperforms state-of-the-art solutions for travel time estimation on the trajectory datasets.

The main contributions are summarized as follows:

1. To our best knowledge, our proposed method is the first time modeling the local congestion with the vehicular flow on target road segments and its neighbors. To that end, we design a GCN module to extract the representation of local congestion, which plays an important role during the model training process.
2. As far as we know, this is the first time that a transformer-based structure is used in the travel time prediction problem, and the relationship between trajectory data is learned through some strategies, such as position coding and multi-head attention.

The rest of this paper is organized as follows. Section 2 introduces the related research work of ETA. Section 3 introduces the proposed TransETA model in details. Section 4 introduces the experimental details. Finally, we conclude this paper in Section 5.

2 Related work

There are many related work for ETA problems, and we will review the previous work on ETA, as well as some work on relevant methodologies.

2.1 Estimated time of arrival (ETA)

Based on statistics and machine learning methods, Jenelius et al. [17] divides travel time into road segment travel time and intersection travel time, and uses statistical models to predict the mean and variance of travel, finally uses the maximum likelihood method to predict the travel time. Hofleitner et al. [18] proposes a probabilistic modeling framework, predicting travel time distributions using sparsely observed probe vehicles and using dynamic bayesian network represents the spatio-temporal dependence on the network. Zhan et al. [19] infers the possible paths for each trip and then estimates the link travel time by minimizing the error between the expected path travel time and observed path travel time. Wang et al. [15] combines with geospatial information, temporal information, and historical contexts learned from trajectories and map data, and fills in the tensor's missing values through a context-aware tensor decomposition approach. Zhang et al. [20] proposes a gradient boosted regression method which combines simple regression trees to predict ETA. The model accounts for spatio-temporal correlations extracted from historical and real-time traffic data. Wang et al. proposed a nearest neighbor-based method is proposed, which estimates the travel time of the current trip by averaging all historical travel times with similar start and end points [21]. However, This non-parametric method is difficult to generalize to situations where no neighbors or the number of neighbors is very limited.

Along with the developments of Neural Networks, ETA methods based on deep learning were widely investigated, as reviewed by Yin et al. [22]. Among the methods based on deep learning, Li et al. [23] proposed a multi-task representation learning model for arrival time estimation (MURAT). This model produces meaningful representation that preserves various trip properties in the real world and at the same time leverages the underlying road network and the spatio-temporal prior knowledge. Wang et al. [24] propose an end-to-end deep learning framework named DeepTTE for travel time estimation. Since the GPS sequence cannot be acquired until the trip is finished, DeepTTE resamples the GPS points by uniform distance at the training stage and generates pseudo points according to the planned route at the inference stage. A multi-layer feedforward neural network named spatio-temporal neural network (ST-NN) for travel time estimation was proposed in paper [25]. ST-NN first uses the discrete latitude and longitude of the starting point and destination as input to predict the travel distance, then the forecast result is combined with time information to estimate travel time.

Reference [26] migrates the wide-deep learning model in the recommender system. The model mainly includes wide module, deep module, and recurrent module. The features are also divided into three parts: continuous features, discrete features and trajectory features. Continuous and discrete features are processed by the wide and deep modules. Trajectory features are processed using the recurrent module. Reference [27] uses GPS data from mobile phones or other probe vehicles. It introduces a method to predict the probability distribution of travel time on an arbitrary route in a road network at an arbitrary time. Reference [28] gives a clear classification of the travel time prediction problem. It is believed that local traffic conditions are closely related to the type of land and building conditions here, and a multi-task end-to-end learning framework was designed to learn travel time. Reference [29] believes that the previous prediction models are all for one vehicle type, such as taxi prediction, and ignore other types of data. Therefore, this paper uses trajectories generated by different types of vehicles to fusion training to predict travel time. Reference [30] combines statistical learning and deep learning, using three hierarchical probability models to predict travel time distribution and reconstruct travel paths, it achieved the best results on multiple datasets. Reference [31] is the travel time prediction model proposed by AutoNavi, which uses the planned traffic flow in the user's travel intention as an approximation of the actual future traffic flow. This method can effectively obtain the flow characteristics, but the data needs to be obtained in real-time and cannot be popularized in real applications. Reference [32] proposes hybrid LSTM and sequential LSTM method based on LSTM neural networks with self-attention mechanism. By introducing self-attention into LSTM, the model is able to capture patterns in location and time sequences in trajectory data.

2.2 Map matching

Map matching refers to the process of correctly projecting the deviated latitude and longitude trajectory points of GPS positioning onto the road to find the true trajectory during travel. As the road network is becoming denser and there are complex scenarios such as parallel and interchanges, it is becoming more and more challenging to accurately projecting trajectory points to the road. According to the sampling range of trajectory points, map matching algorithms are mainly divided into two categories: local-based algorithm and global-based algorithm. The local-based algorithm uses a greedy strategy to expand the solution sequentially from the matched part and try to find the local optimal point based on the distance and direction similarity [33, 34]. The local-based method is very efficient and is usually used in online applications. But this method reduces the matching accuracy when the sampling rate of the trajectory is low.

The goal of the global-based algorithm is to match the entire trajectory with the road network considering both the previous points and subsequent points. The global-based algorithm is more accurate than the local-based method usually used for offline tasks such as mining frequent trajectory patterns, but its efficiency is relatively low due to the entire trajectory needs to be generated [35, 36]. Later, reference [37] proposed to use both local and global information to process the sparse trajectory points, the algorithm first finds the local candidate road segments within one circle of each point in the trajectory and then processes it.

2.3 Graph convenlutional network (GCN)

At present, GCN is mainly divided into two categories, one is spectral convolution, which performs convolution transformation in the Fourier domain; the other is non-spectral convolution, which performs convolution directly on the graph. For spectral convolution, there are three main progresses. In reference [38], the author proposed the relationship between convolution and Fourier transformation. First, the product is multiplied in the Fourier domain and then the inverse Fourier transform is performed. Reference [39] targets the three problems of reference [38] that each convolution operation requires matrix multiplication and does not consider spatial locality and all nodes must be considered for each convolution. Reference [40], compared with the method in reference [38], has made further improvements, that is the depth of the model is deepened and the width is reduced.

For non-spectral convolution, which is also called spatial domain convolution, a lot of work has also been proposed. Reference [41] proposed diffusion-convolutional neural networks (DCNN), which is mainly used for node classification and graph classification tasks. Reference [42] proposed for graph classification tasks, select some nodes to represent the entire graph and select a specific number of neighborhoods for each node, then convolve on the matrix composed of each node and its neighborhood nodes. By incorporating the attention mechanism in [43], the correlation between nodes can be dynamically calculated, and the model can be learned directly or inductively.

On the whole, spectral convolution has a complete set of theoretical support, but sometimes it is limited by the laplacian operator; non-spectral convolution is more flexible, and the difficulty is how to choose the right quantification area.

3 Methodology

3.1 Notation

In order to describe our approach, we first define some variables.

T	the trajectory of trip refers to a sequence of geographical GPS points.
p_i	GPS points in trip.
$p_{i \cdot lon}$	longitude of p_i.
$p_{i \cdot lat}$	latitude of p_i.
t_i	timestamp of the i-th GPS point.
t_p	the travel time of a path(query q).
o_q	the origin of query q.
d_q	the destination of query q.
s_q	the departure time of query q.

3.2 Preliminary

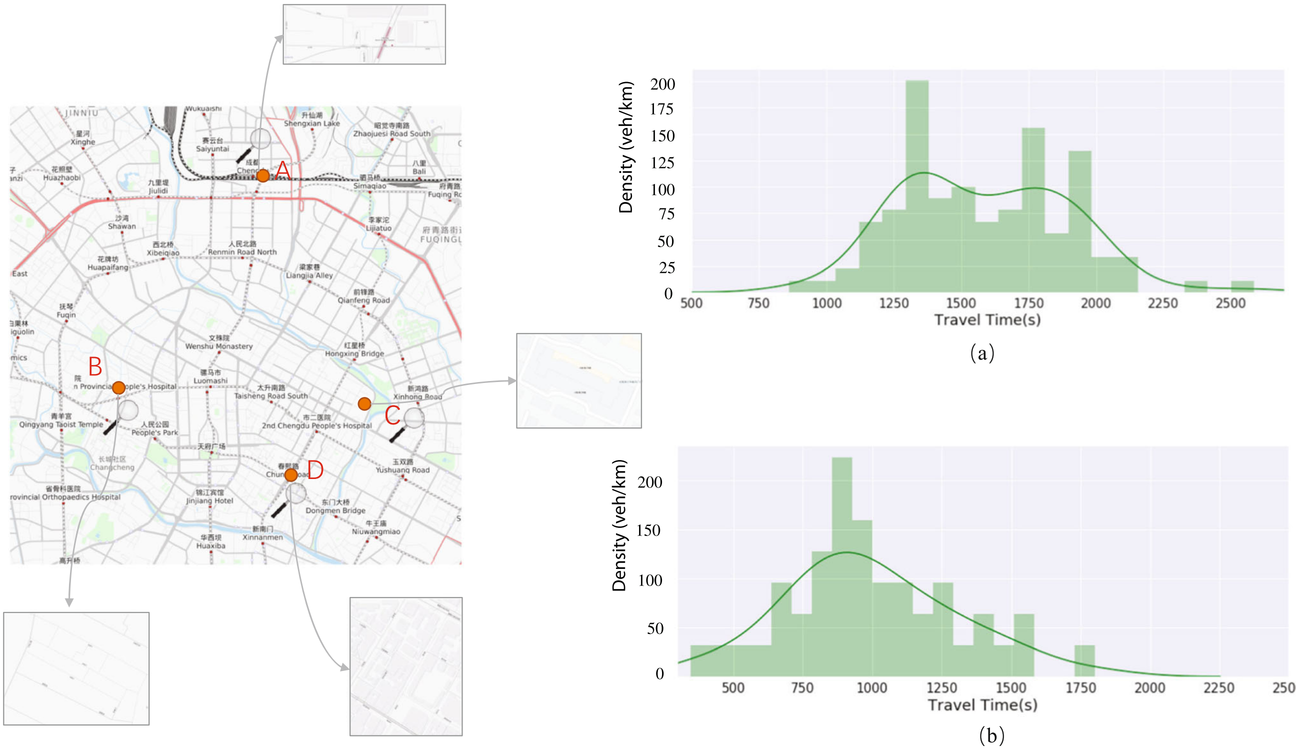
Definition 1 Trajectory The trajectory T of trip refers to a sequence of geographical GPS points. That is, T = $\{p_1, p_2 ... p_n\}$, where n is the number of GPS points in trip, GPS points p_i contains longitude $p_i \cdot_{lon}$, latitude $p_i \cdot_{lat}$ and timestamps t_i. Furthermore, for each trajectory we record its external factors such as the starting time (Time ID), the vehicle state (State ID), the weather condition (weather ID) and corresponding driver (Driver ID).

Definition 2 Travel Time The travel time of a path is defined as $t_p = t_n - t_1$. The travel distance of the trip equals the accumulation of great-circle distances between two consecutive footprints. Therefore, given a query q = (o_q, d_q, s_q), our goal is to estimate the travel time t_p with a given origin o_q ,destination d_q, departure time s_q, and the external features.

3.3 Data preprocessing

Vehicle trajectory data usually consists of vehicular id, departure time, and trajectory information. In the data processing part, we divide the data into two categories: static data and driving data. The driving data refers to the positions and the corresponding timestamps that the vehicle passes through during the driving process; static data is the data describing the travel of the vehicle except for driving information, which does not change over time. As shown in Fig. 1, we calculated the distributions of travel time between two pairs of selected regions in Chengdu. Area A is the Chengdu railway station, area B is the Kuanzhai Alley scenic area in Chengdu, area C is a residential area, and area D is the Chunxi Road commercial street. Figure 1(a) presents the travel time distribution from A to B, and Fig. 1(b) presents the travel time distribution from C to D. It can be found that the travel time distribution between the same two regions is very scattered, and the prediction of future travel time based solely on the data distribution is much challenging.

Specifically, we present the data processing procedure as follows as follows: for static data, the data with missing values is first deduplicated to reduce the impact of noise data; then the original time features are split into years, months, weeks of year, days of year, hours of day, and minutes of hour; the split time features and other static features are used as input; For the driving data, the map-matching method is used to map the data points to the road networks, then the number of vehicles on each road segment in each period is counted as the proxy for the traffic flow. During the original data collection process, we sampled the data at equal time intervals. If the samplings are used directly, the model can calculate the travel time by calculating the number of sample points along the route, which may make the model overfitting. Therefore, we re-sample the data at equal distances



(a)

(b)

Fig. 1 Two examples of travel time distribution.(a) Travel time distribution from area A to area B.(b) Travel time distribution from area C to area D

and calculate the time of the corresponding position through linear interpolation. For the time of the trajectory points, we also embed them as a part of inputs.

3.4 TransETA

The network structure of the proposed TransETA is shown in Fig. 2, which is mainly composed of input feature transformation module, ETA-Transformer module, and deep forest module. The input feature transformation module uses GCN to extract road flow characteristics; the deep forest module processes static data; the ETA-Transformer module mainly processing feature extraction on dynamic trajectory data.

3.4.1 Input feature transformation

For traffic networks, the traffic flows allocate only in the network of traffic roads, and the rest area of the city may not need to be considered. Taking the road network topology into consideration, GCN is applied to extract the traffic flow features. The graph structure of GCN is very consistent with the road network structure, and it has enough ability to extract the features of traffic flow networks. The road flow data obtained from data preprocessing is used as the input of GCN. We abstract each road segment as a node in the graph. If two road segments are connected, then there is an edge connecting between the two nodes. The flow of this road segment in the current period is used as the current feature of the node. Figure 3 depicts an example of road flow at 8 am and 3 pm. The shade of the color indicates the traffic flow at the current moment. For each node, we take its neighboring nodes' information to update the representation of this node. Through the GCN module, the traffic information of the road segment passes through the 5 layers GCN unit. The vector learned from the current road segment is used as an input of the ETA-Transformer module, as shown in Fig. 4.

The driving data has both periodic and non-periodic characteristics. During the processing, the road flow stays unchanged within the selected time interval, and will not vibrate due to randomness. Based on these features, it is suitable to process the time information of each trajectory point in the driving data with Time2Vec. Compared with the traditional method of designing time windows, Time2Vec is simple and convenient and does not require too much domain knowledge. The model can effectively convert dynamic timing into static vectors for subsequent applications. The calculation principle of Time2Vec is described as follows:

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \varphi_i , & \text{if } i = 0. \\ F\left(\omega_i \tau + \varphi_i\right) , & \text{if } 1 \leq i \leq k \end{cases} \tag{1}$$

where τ represents the original features of the time series, φ and ω are the weight coefficients to be learned, F is the periodic activation function, and the sine function k is usually used to represent the Time2Vec dimension. Periodic mode is captured when i equals 0, and non-periodic mode is captured

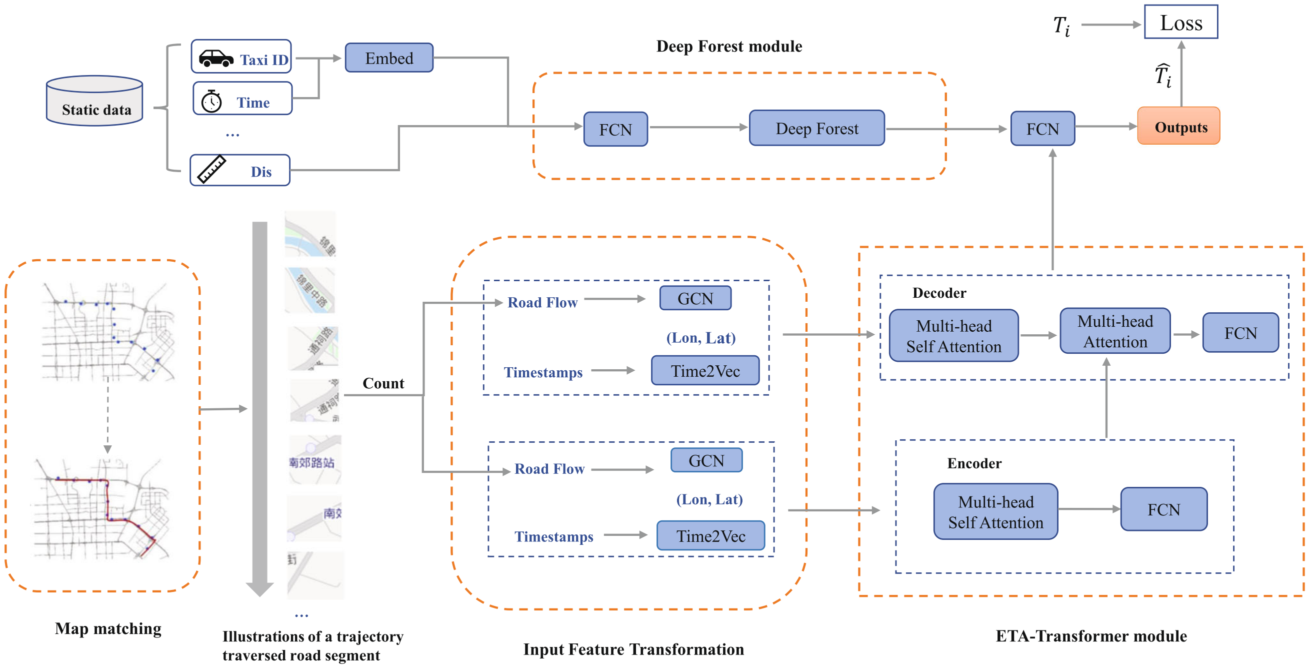


Fig. 2 The overall architecture of TransETA. The model is composed of three parts: Input feature transformation module, ETA-Transformer module, and Deep Forest module

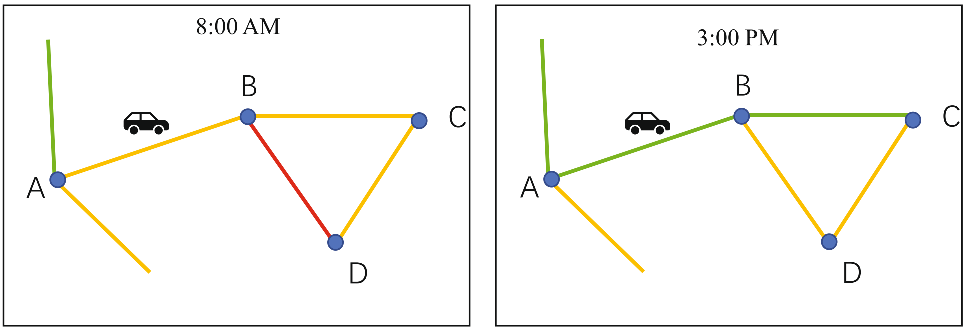


Fig. 3 The road flow changes at 8 a.m. and 3 p.m., the shade of the color indicates the amount of traffic

when i in the range $[1, k]$. The vector processed by Time2Vec is also used as an input of the ETA-Transformer module.

3.4.2 ETA-transformer module

The ETA-Transformer module is shown in the right part of Fig. 2, which is divided into two parts: encoder and decoder. Compared with the LSTM-based network, the transformer can directly capture the long-distance dependence in the sequence and effectively reduce the training time through the parallel strategy.

In the encoder module, we use 5 identical units. Each unit includes two components: multi-head self attention and FCN. The transformed trajectory data is input into multi-head self-attention. Previously extracting sequence features mainly used CNN and RNN modules, and CNN-based method assumes that local information depends on each other. Local feature information is extracted from the data according to the size of the convolution kernel, but global information cannot be extracted; The core of the RNN-based method is long-distance dependent and keep the information flowing in the process of dissemination, however since the information flow must be executed sequentially from the beginning to the end, it cannot be parallel processed. Multi-kernel method in CNN can consider larger spatial information on the same layer. The self-attention method considers both the spatial and temporal traffic information at the same time. Similar to the multi-kernel method in CNN, multi-head attention can learn not only the auto correlation of the historical data, but also the cross correlation of the spatial data. The three mechanisms are illustrated shown in Fig. 5.

We use multi-head self-attention to learn the trajectory data information output by the input feature module, where Q, K, V are all linear mappings to the input X, d_k is the dimension of the key vector.

$$\text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

A very important innovation of transformer is the use of position encoding, which encodes different positions to a certain extent. The obtained vector X with information of driving data is linearly encoded differently in different positions, and we derive matrices of Q, K, and V as follows:

$$Q = X \cdot W_i^Q \tag{3}$$

$$K = X \cdot W_i^K \tag{4}$$

$$V = X \cdot W_i^V \tag{5}$$

where Q, K, and V focus on different positions of the input vector X, that is having different attentions on the spatial and temporal information of road flows and timestamps. Transformers uses the attention mechanism to construct the features of each vector of traffic driving information, so as to find out the important parts of the traffic information, and build up the relevance of the vector, thus to extract the correlation of the spatial and temporal information of traffic

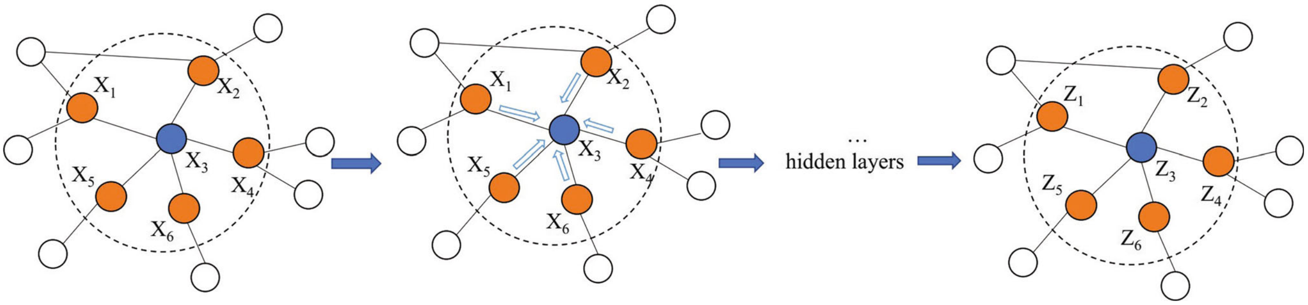


Fig. 4 Use GCN to learn road flow characteristics

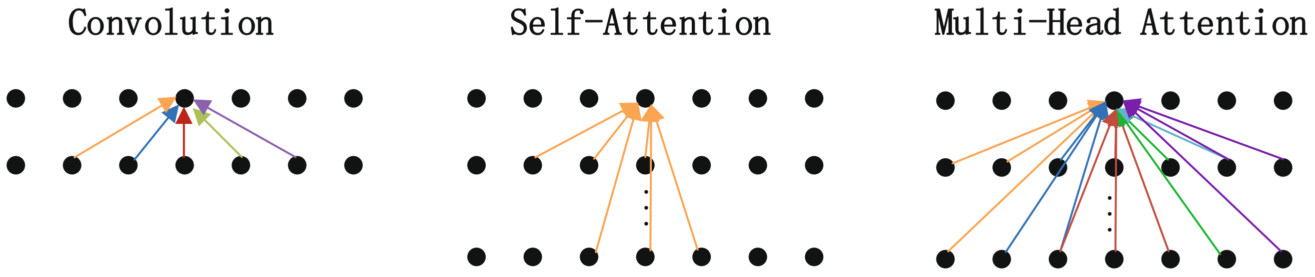


Fig. 5 Comparing CNN,Self-Attention and Multi-Head Attention

networks. For head i of multi-head, first processed by linear transformation using matrix W_i and then go through Attention.

$$\tilde{V}_i = \text{Attention}(Q_i, K_i, V_i) \tag{6}$$

Next, we concatenate multiple head i, using linear transformation via matrix W^o to get the final output of the multi-head. The overall feature obtained with different attentions is only the sum of linear transformations of all traffic data features, which are weighted according to their importance or focuses.

$$\tilde{V}_m = \text{MultiHead}(Q, K, V) = \text{Concat}(\tilde{V}_1, ..., \tilde{V}_a)W^o \tag{7}$$

The output of multi-head self-attention first process by ReLU activation function and then input into the fully connected network FCN.

$$o_m = W_2 \max\left(0, W_1 \tilde{V}_m + b_1\right) + b_2 \tag{8}$$

where $\max(0, \cdot)$ is the ReLU activation function, W_1 and b_1 are the linear parameters for the output of multi-head self-attention, W_2 and b_2 are the parameters for the FCN. Finally, the encoded vector is input to the multi-head attention part of the decoder.

The decoder part is similar to the encoder. Five identical units are used in the model. Each unit has one more multi-head attention than the encoder to build the correlation between the output of the encoder and the decoder. Compared with the encoder, the multi-head self-attention value after calculating position i is set to $-inf$. Because the attention of i position cannot depend on the sequence after i. The final output vector of the decoder will determine the final result together with the output of the deep forest module.

3.4.3 Deep forest module

Deep forest [44] is a tree-based ensemble model. In [44], it believes the effectiveness of deep learning is mainly attributed to three strategies: multi-layer information processing, feature conversion, and sufficient complexity. However, the method based on deep learning have numerous parameters. So, deep forest tries to overcome the drawbacks of deep learning, which not only has an effective strategy as deep learning, but also solves some of the shortcomings of deep learning. The model mainly includes two parts: multi-grained scanning and cascade forest. Since deep forest is good at dealing with different data from different domains and is robust to hyper-parameter settings, it is selected to learn the features of static traffic data in urban traffic networks.

The deep forest is mainly used to process static data here. First, we input static data to the multi-grained scanning part. For sequential data, assuming that the input feature has 400 dimensions, a sliding window with a size of 100 is used to extract the feature by sliding on the original data. After the single-step sliding is completed, 301 feature vectors are obtained. For spatial data, such as a picture with a size of 20*20, 121 feature vectors are obtained after using a sliding window of 10*10, then the features input into the multi-granularity forest. There are two types of forest: one is a random forest, and the other is a complete random forest. The difference lies in whether the split feature and threshold are randomly selected. Finally, cascade the output of the vector by the two forests to get the output.

The cascading forest is a hierarchical sequential structure, the output of the previous layer is the input of the next layer. The output of the 4 forests and the original feature vector of the input are cascaded as the input of the next layer. Finally, the output vector is obtained by averaging the output of the 4 forests in the last layer.

4 Experiments

4.1 Data description

(1) Chengdu dataset: The data includes 9,737,557 trajectories of 14,864 Chengdu taxis in August 2014. The shortest trajectory has only 11 trajectory points (2km). The longest trajectory includes 128 trajectory points (41km). The data is sampled at 60-second intervals.

(2) Porto dataset: The data includes the trajectories of 442 taxis in Porto, Portugal, from July 1, 2013 to June 30, 2014. The trajectory data is sampled at 15-second intervals.

The two trajectory datasets are different in some static features. The Chengdu dataset has features such as taxi ID and passenger status, while the Porto dataset has features such as travel ID and date type. The processing method is to embed the discrete feature into the vector and then use it as the subsequent input. The statistical information of the datasets is shown in Table 1. The study area and part distribution of raw GPS data are presented in Figs. 6 and 7.

4.2 Baseline methods

In the experiment, TransETA is compared with the following methods:

1. XGBoost: XGBoost is currently the most popular tree-based ensemble learning model. The method based on XGBoost is widely used in data competitions and industry because of its excellent result and fast training speed [10].
2. LSTM: LSTM is currently the most popular method for processing sequential data [13]. A pieces of work has used LSTM to estimate the travel time in freeways and urban road networks [45].
3. TEMP: This method estimates the travel time of the current trajectory by averaging all historical travel times with similar starting point and destination. Calculate the scale factor based on the relative time speed reference value in the time change of the average speed of all trips in the city [21].
4. DeepTTE: A recently proposed end-to-end deep learning prediction method. The model in the paper is divided into three parts: spatio-temporal module, attribute module, and multi-task learning module. The time and space dependence of the original trajectory point is learned through the trajectory information [24].
5. DeepGTT: Combining statistical learning and deep learning together, DeepGTT uses three hierarchical probability models to predict travel time distribution and reconstruct travel paths [30].

Table 1 The statical information of the datasets

Dataset	Chengdu	Porto
# vehicles	14864	442
Average travel time	1282s	749s
Travel time std	1612s	593s
Average travel distance	5.98km	5.79km
# days for training	21	273
# days for testing	7	90

4.3 Experimental settings

In the static feature input module, we embed the “month” feature as a 4-dimensional vector, the “day” feature as a 10-dimensional vector, the “hour” as an 8-dimensional vector, “minute” is embedded as a 15-dimensional vector. “driver ID” is embedded as an 8-dimensional vector and for the second-class features of “passenger carrying status” using one-hot encoding it.

In the deep forest module, we set the number of bins to 255, i.e., n_bins = 255, the number of samples per bin to 200000, i.e., bin_subsample = 2e5, the maximum number of cascading layers in depth in the deep forest to 20, i.e., max_layers = 20, the number of estimators per cascading layer to 3, i.e., n_estimators = 3, and the number of trees per estimator to 100, i.e., n_trees = 100. In the TransETA module, 8 heads are used. For each heads query vector and key vector dimension $d_k = 64$, value vector $d_v = 64$. The output dimension in the fully connected layer is $d = 512$. For the training process, we use Adam optimizer, $learning\ rate =$ 1e-3, $batch\ size = 64$, $units\_num = 5$, and the learning rate is reduced by 2 times after every 2 epochs. To obtain converged results, the number of iterations for model training varies for individual models and different datasets. All methods use python 3.6.4 environment and PyTorch 1.2.0 framework to train on GeForce RTX 2080Ti.

We use mean absolute error (MAE) and mean absolute percentage error (MAPE) to evaluate the performance of the model.

$$MAE(T,\hat{T}) = \frac{1}{N}\sum_{i=1}^{N}\|T_i - \hat{T}_i\| \tag{9}$$

$$MAPE(T,\hat{T}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\|T_i - \hat{T}_i\|}{T_i} \times 100\% \tag{10}$$

Among them, T_i represents the real travel time, $\hat{T}_i$ represents the predicted value of the model, and N represents the number of trajectories in the test set.

4.4 Analysis of results

The experimental results are shown in the Table 2, and the best experimental results are highlighted in bold. It can be found that our method achieved the best performance on the two datasets of Chengdu and Porto. Compared with

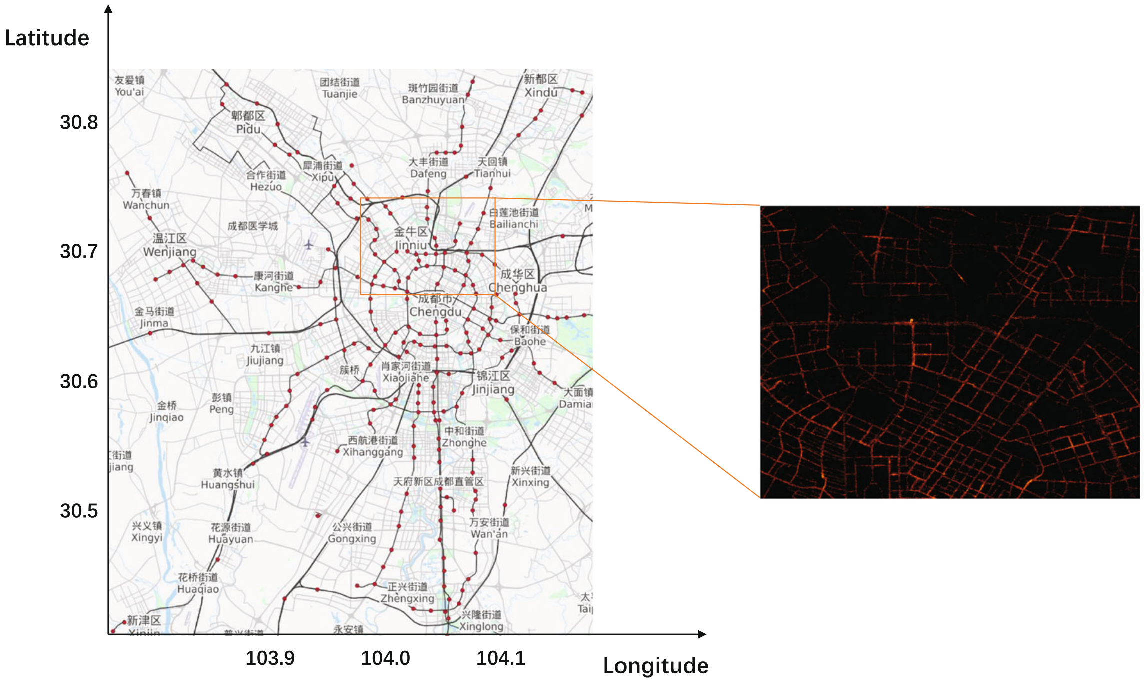


Fig. 6 Region and spatial distribution of GPS footprints in Chengdu

the TEMP, XGBoost, and LSTM methods, both MAE and MAPE metrics have dropped dramatically. Among the baseline models on the Chengdu dataset, DeepTTE achieved the best results. Compared with DeepTTE, our method reduces MAE by 6.19 seconds and MAPE by 2.34%; Among the baseline models on the Porto dataset, DeepGTT achieved the best results. Compared with DeepGTT, our method reduces MAE by 8.73 seconds and MAPE by 3.64%.

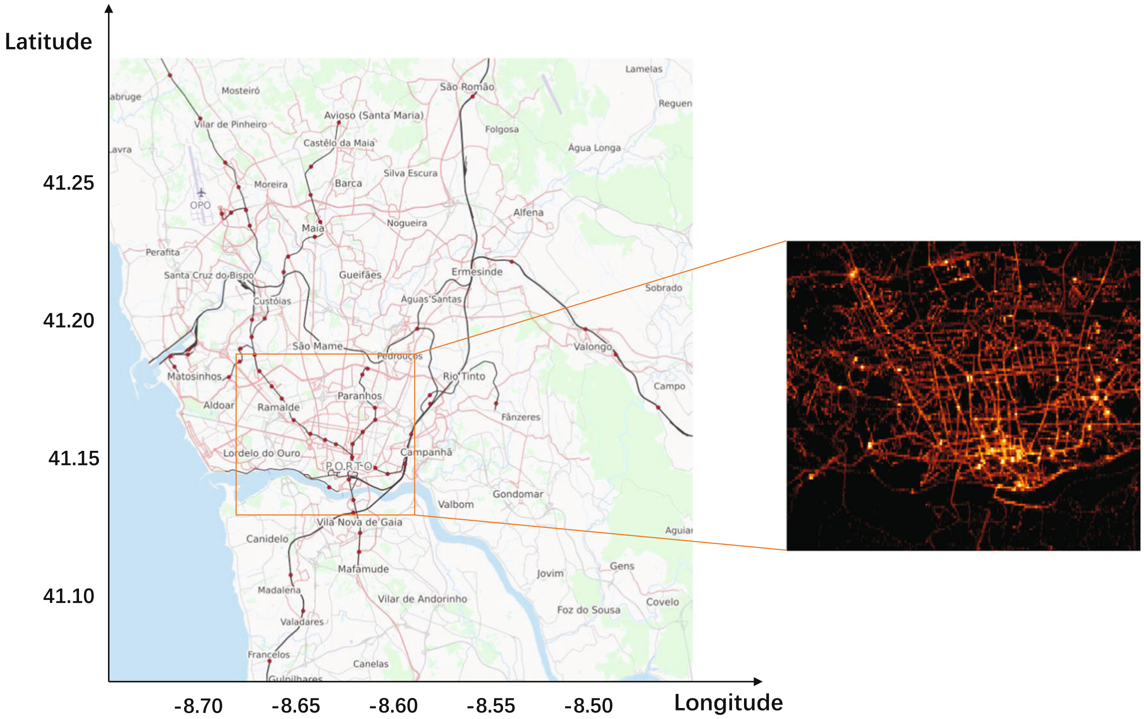


Fig. 7 Region and spatial distribution of GPS footprints in Proto

Table 2 The comparison result on Chengdu and Porto Dataset

Dataset	Metrics	TEMP	XGBoost	LSTM	DeepTTE	DeepGTT	TransETA
Chengdu	MAE(s)	284.21	254.09	278.50	190.53	195.08	**184.34**(↓**6.19**)
	MAPE(%)	22.87	18.51	21.73	13.07	15.91	**10.73** (↓**2.34**)
Porto	MAE(s)	235.89	213.62	242.16	148.26	143.65	**134.92** (↓**8.73**)
	MAPE(%)	28.21	25.76	30.15	20.15	19.78	**16.14**(↓**3.64**)

The bold entries is used to emphasize that our proposed method has the best performance compared to other methods

10 trips were randomly selected from the Chengdu dataset, and the predicted and actual values are shown in Fig. 8. The horizontal axis of the image represents the sampling points, the vertical axis represents the travel time in seconds, and the number on top of the bar graph represents the absolute value of the difference between the true value and the predicted value. From trip 1 to trip 10, they are sorted according to the travel time from small to large. We can observe that with the increase of travel time, the error shows an increasing trend, which indicates that with the increase of travel time, the uncertain factors in the vehicle operation process increase, and the difficulty of forecasting gradually increases.

To verify the effectiveness of each module of the proposed model, we conducted ablation experiments on TransETA to verify the effectiveness of each module.

(1) w/o GCN: Removed GCN to extract road flow information part, the input feature transformation module only contains trajectory longitude and latitude data and trajectory time data processed by Time2Vec.

(2) w/o DF: Using multi-layer perceptron instead of deep forest module to process static data.

(3) w/o Tr: Using LSTM instead of ETA-Transformer module to process trajectory data.

The experimental results are shown in the Table 3. After removing the GCN module, the MAE and MAPE metrics significantly rise, indicating that the GCN learned the changes in road traffic flow. For example, on the Chengdu dataset, our proposed ETA-Transformer method achieved an MAE of 184.34s. After removing the GCN feature extraction module, the MAE increased to 192.15, indicating that adding GCN improves the performance of our model. In addition, we observed that removing the GCN module resulted in a more significant performance drop compared to the other two modules, suggesting that feature extraction is crucial for time series prediction. The addition of road traffic flow information with the framework of GCN significantly improved the prediction results. After the deep forest module in w/o deep forest is replaced by MLP, the performance of the adapted method decreases, indicating that deep forest can better learn static feature information compared with MLP. This is because deep forest is good at dealing with different data from different domains and is robust to hyper-parameter settings. By substituting the ETA-Transformer module with LSTM, the ETA-Transformer module can better find the correlation between the spatial and temporal traffic network data, and thus has a better prediction performance. The results demonstrate that the three components of GCN, deep forest and transformer all improved performance of TransETA model.

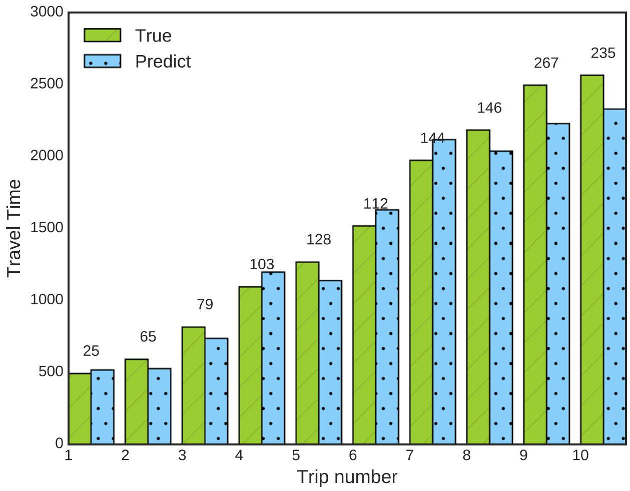


Fig. 8 Predicted and actual values of 10 trips in Chengdu dataset

Further, we compare the performance of DeepTTE, DeepGTT, and TransETA from multiple dimensions on the two datasets. Figures 9 and 10 present the MAPE change with different travel time and travel distance respectively, We observe that: (1) under different travel time and distance, the performance of the TransETA model is better than DeepTTE and DeepGT; (2) with the increase of travel time, the prediction errors of the three models all increase to varying degrees, but TransETA is more stable than the other two methods, while DeepTTE and DeepGTT perform poorly on long-term trips; with the travel distance increases, the MAPE shows a downward trend, which indicates that the increase in the distance leads to a gradual increase in the absolute error of the forecast but a gradual decrease in the relative error; (3) Chengdu and Proto have different sensitivity to different travel times and distance. Compared with the Proto dataset, the Chengdu dataset fluctuates more in different periods and Proto changes slightly.

Figure 11 presents the estimation error per hour, all models show relatively weak performance during the peak hours due

Table 3 Comparison of TransETA ablation experiment results

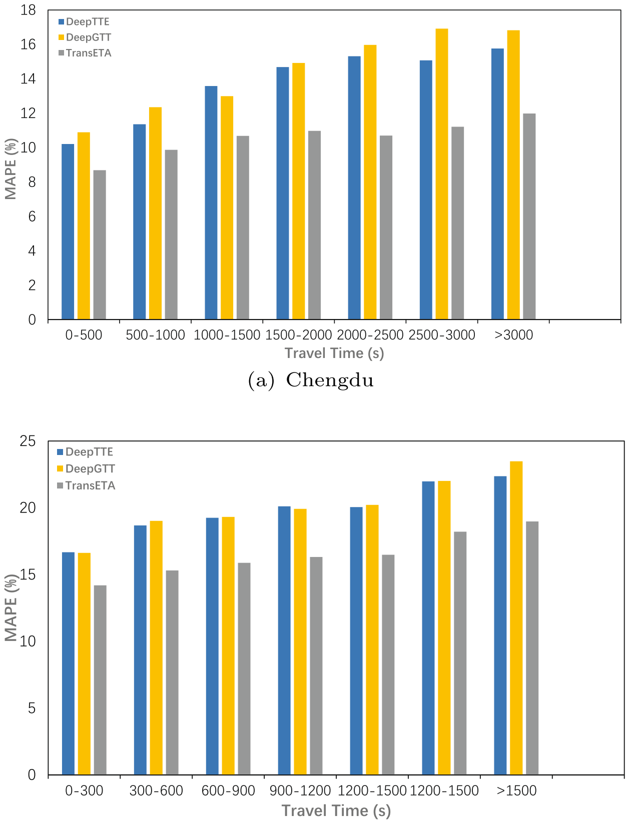
Dataset	Metrics	w/o GCN	w/o DF	w/o Tr	TransETA
Chengdu	MAE(s)	192.15	187.39	189.64	**184.34**
	MAPE(%)	14.62	11.93	12.55	**10.73**
Porto	MAE(s)	144.33	137.47	140.52	**134.92**
	MAPE(%)	19.97	17.91	19.06	**16.14**

The bold entries is used to emphasize that our proposed method has the best performance compared to other methods

to the traffic congestion in the Chengdu dataset. Even so, TransETA generates better estimation than compared models during all the day. In the Proto dataset, the performance is stable in all periods.

We also analyzed the selection of hyperparameters in the model, the hyperparameter performance heat map of Chengdu dataset and Proto dataset are shown in Figs. 12 and 13 respectively. *max_layers* and *n_estimator* are the main parameters of the deep forest module, *learning_rate* and *units_num* are the main parameters of the TransETA module. Grid search is performed on the parameter candidate set to find the best combination of parameters.

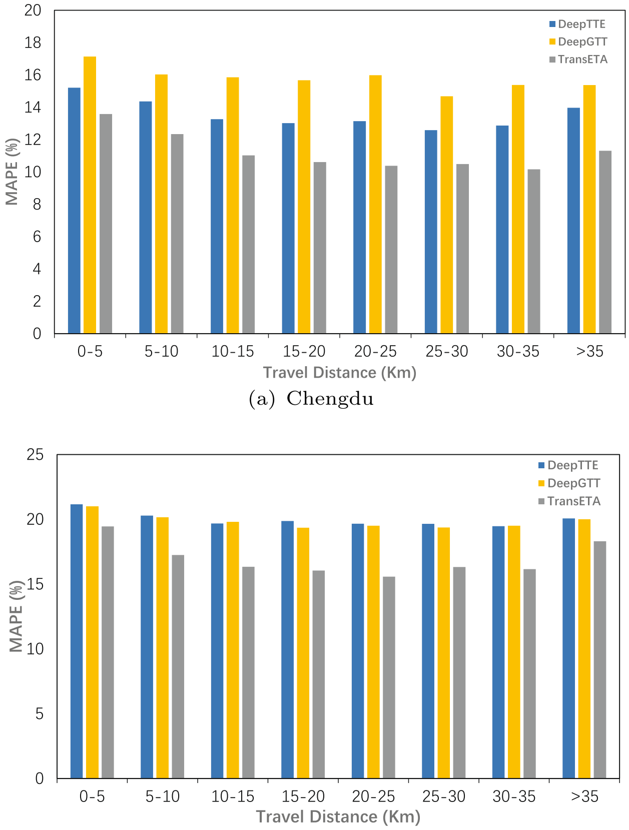
In terms of data processing, the static and dynamic data of road networks are divided, and graph convolutional neural networks (GCN) are employed to extract temporal and spatial information from the road network. To prevent overfitting on the time series data, an equidistant resampling technique is applied, where linear interpolation is used to compute the corresponding time for each sample. The model incorporates Transformers to accelerate training and effectively handle long-term dependencies. Therefore, theoretically, the proposed method can provide accurate point-to-point travel time prediction. However, in practice, training such a model requires a large amount of road network data, and the road network conditions vary across different cities, thus when the model is applied to a new city, the algorithm needs to be adapted based on new data.



(a) Chengdu

(b) Proto

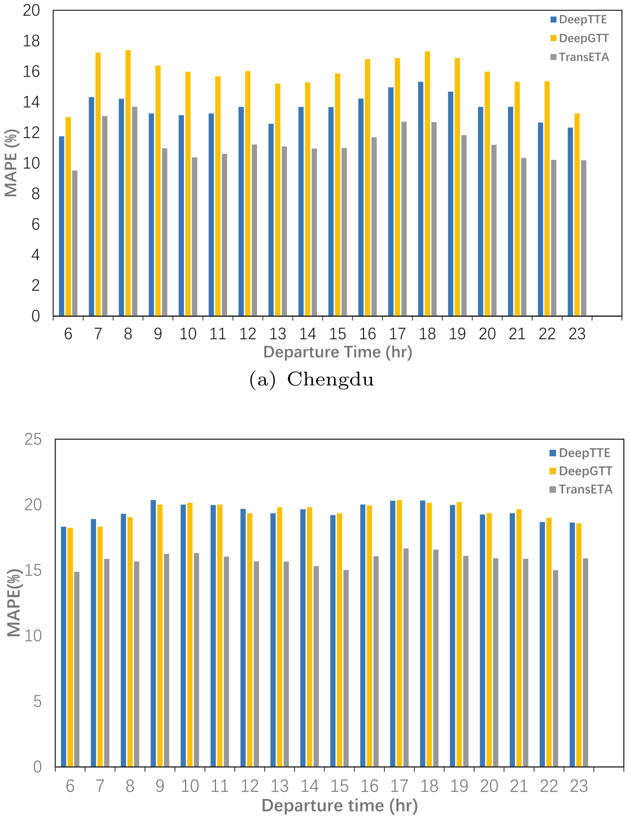
Fig. 9 Changes of MAPE for trajectories with different **travel time**. (a) Chengdu (b) Proto



(a) Chengdu

(b) Proto

Fig. 10 Changes of MAPE for trajectories with different **travel distance**. (a) Chengdu (b) Proto



(a) Chengdu

(b) Proto

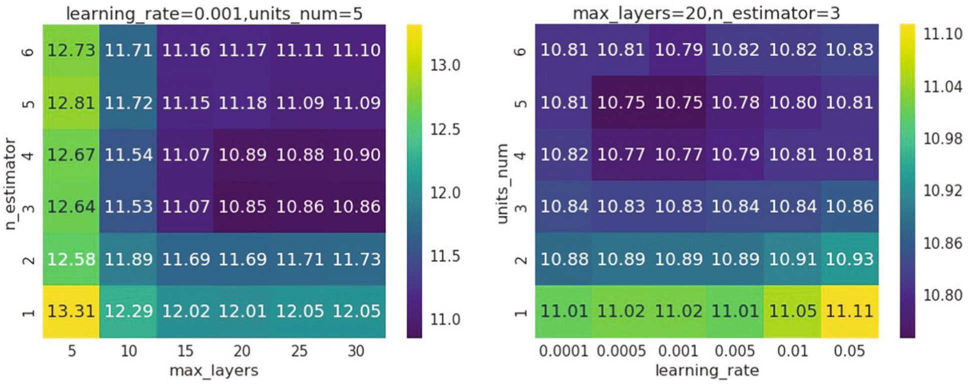
Fig. 11 Changes of MAPE for trajectories with different **departure time**. (a) Chengdu (b) Proto

5 Conclusion

In this work, we propose a new model TransETA to predict the travel time of vehicles. The model consists of three modules: the input feature transformation module uses GCN to extract road flow characteristics, the deep forest module mainly processes static data, and the ETA-Transformer module mainly processing feature extraction on dynamic trajectory data. The utilization of graph convolutional neural networks (GCN) enables the extraction of state information from the road network. As road networks possess inherent graph structures, GCNs can effectively extract features. Transformers are capable of capturing long-term dependencies in time series more efficiently and can reduce training time through parallel strategies. For data processing, the adoption of map-matching techniques maps data points to the actual map and subsequently performs equidistant resampling. Linear interpolation is applied to calculate corresponding time for the resampled data, thereby reducing the extent of model overfitting. Then, we conducted experiments on the Chengdu and Porto trajectory datasets, the experimental results show that the proposed method is better than other travel time prediction baselines, where the reduction of MAE(s) reached up to 8.73%. Although, it is compared with the state-of-the-art methods, the proposed method still has non-negligible improvement for the performance. Finally, the ablation experiment shows the effectiveness of each module, the addition of road traffic flow features can effectively improve the accuracy of prediction; Compared with the method based on LSTM, the ETA-Transformer module can effectively extract the correlation of features. Travel time prediction can be used for taxi service, ride-sharing service, emerging cases, like ambulance and fire trucks.

Even though comparatively good results are obtained in this work, better prediction can always be achieved by taking into consideration of more detailed travel information. Moreover, significant parameter tuning effort is required during the experimental training process. Therefore, in the future, we intend to improve this work in two aspects: (i) Multi-source data fusion. The data currently used are mainly tabular data, such as trajectory and road network data. Image and video data will be further integrated to enrich the dimensions of the data, the performance of the model will be improved through multi-modal data. In addition, we would like to take into consideration of more traffic information to further improve the prediction accuracy, such as resident info, travel habits, travel behavior, etc. (ii) The addition of hyperparameter search algorithm. In the process of network design, parameter adjustment is time-consuming. Later, we will consider introducing hyperparameter search and network architecture

Fig. 12 Hyperparameter performance heat map of Chengdu dataset



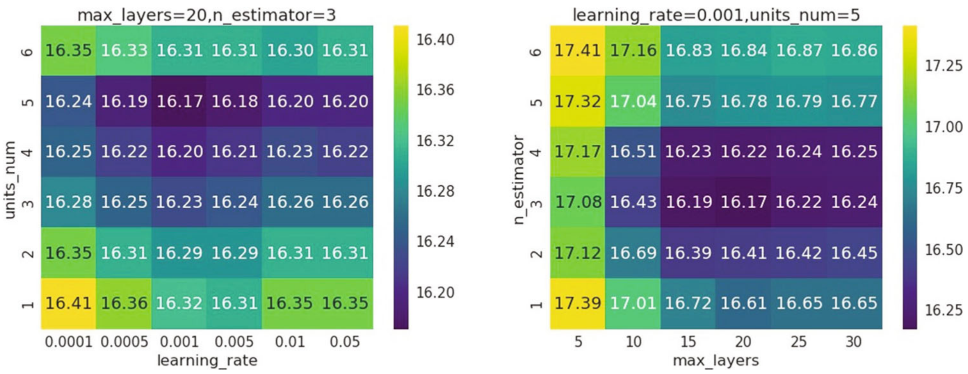


Fig. 13 Hyperparameter performance heat map of Proto dataset

search method into the model to automatically find the best network structure and the best combination of parameters.

Acknowledgements The research is supported by the National Key R&D Program of China (2018YFB1600500), the National Science Foundation of China (61673366, 61620106009, 62102258), the European COST Action TU1102, the Shanghai Pujiang Program (21PJ1407300) and the Fundamental Research Funds for the Central Universities. We appreciate the valuable insights and significant contributions provided by Hu Hui Feng in the paper revision.

Author Contributions All the authors contribute equally to the paper.

Funding Not applicable

Availability of data and materials Available upon reasonable request.

Code Availibility Available upon reasonable request.

Declarations

Ethics approval Not applicable

Consent to participate Not applicable

Consent for publication Not applicable

Conflict of interest No potential conflict of interest was reported by the authors.

References

1. Gers FA, Eck D, Schmidhuber J (2002) Applying lstm to time series predictable through time-window approaches, 193–200
2. Zhang GP (2003) Time series forecasting using a hybrid arima and neural network model. Neurocomputing 50:159–175
3. Contreras J, Espinola R, Nogales FJ, Conejo AJ (2003) Arima models to predict next-day electricity prices. IEEE Trans Power Syst 18(3):1014–1020
4. Akaike H (1969) Fitting autoregressive models for prediction. Ann Inst Stat Math 21(1):243–247
5. Holt CC (2004) Forecasting seasonals and trends by exponentially weighted moving averages. Int J Forecast 20(1):5–10
6. Gustafsson B, Kreiss H-O, Oliger J (1995) Time dependent problems and difference methods, 24
7. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory, pp 144–152
8. Quinlan JR (2014) C4. 5: programs for machine learning
9. Qiu J, Wu Q, Ding G, Xu Y (2016) Feng S (2016) A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing 1:1–16
10. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining, pp 785–794
11. Ruck DW, Rogers SK, Kabrisky M (1990) Feature selection using a multilayer perceptron. Journal of Neural Network Computing 2(2):40–48
12. Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211
13. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
14. Yang B, Guo C, Jensen CS (2013) Travel cost inference from sparse, spatio temporally correlated time series using markov models. Proceedings of the VLDB Endowment 6(9):769–780
15. Wang Y, Zheng Y, Xue Y (2014) Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 25–34
16. Babaei A, Khedmati M, Jokar MRA, Tirkolaee EB (2023) Sustainable transportation planning considering traffic congestion and uncertain conditions. Expert Syst Appl 227:119792. https://doi.org/10.1016/j.eswa.2023.119792
17. Jenelius E, Koutsopoulos HN (2013) Travel time estimation for urban road networks using low frequency probe vehicle data. Transport Res Part B: Methodological 53:64–81
18. Hofleitner A, Herring R, Abbeel P, Bayen A (2012) Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. IEEE Trans Intell Transp Syst 13(4):1679–1693
19. Zhan X, Hasan S, Ukkusuri SV, Kamga C (2013) Urban link travel time estimation using large-scale taxi data with partial information. Transport Res Part C: Emerg Technol 33:37–49
20. Zhang F, Zhu X, Hu T, Guo W, Chen C, Liu L (2016) Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations. ISPRS Int J Geo Inf 5(11):201
21. Wang H, Kuo YH, Kifer D, Li Z (2016) A simple baseline for travel time estimation using large-scale trip data. In: 24th ACM SIGSPATIAL International conference on advances in geographic information systems, ACM SIGSPATIAL GIS 2016, p 61. Association for Computing Machinery
22. Yin X, Wu G, Wei J, Shen Y, Qi H, Yin B (2022) Deep learning on traffic prediction: methods, analysis, and future directions. IEEE Trans Intell Transp Syst 23(6):4927–4943

23. Li Y, Fu K, Wang Z, Shahabi C, Ye J, Liu Y (2018) Multi-task representation learning for travel time estimation. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1695–1704
24. Wang D, Zhang J, Cao W, Li J, Zheng Y (2018) When will you arrive? estimating travel time based on deep neural networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
25. Jindal I, Chen X, Nokleby M, Ye J et al (2017) A unified neural network approach for estimating travel time and distance for a taxi trip. arXiv:1710.04350
26. Wang Z, Fu K, Ye J (2018) Learning to estimate the travel time. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 858–866
27. Woodard D, Nogin G, Koch P, Racz D, Goldszmidt M, Horvitz E (2017) Predicting travel time reliability using mobile phone gps data. Transport Res Part C: Emerg Technol 75:30–44
28. Lan W, Xu Y, Zhao B (2019) Travel time estimation without road networks: an urban morphological layout representation approach. IJCAI, 1772–1778
29. Lin X, Wang Y, Xiao X, Li Z, Bhowmick SS (2019) Path travel time estimation using attribute-related hybrid trajectories network. In: Proceedings of the 28th ACM international conference on information and knowledge management, pp 1973–1982
30. Li X, Cong G, Sun A, Cheng Y (2019) Learning travel time distributions with deep generative model. In: The World Wide Web conference, pp 1017–1027
31. Dai R, Xu S, Gu Q, Ji C, Liu K (2020) Hybrid spatio-temporal graph convolutional network: improving traffic prediction with navigation data. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 3074–3082
32. Sun J, Kim J (2021) Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks. Transport Res Part C: Emerg Technol 128:103–114
33. Civilis A, Jensen CS, Pakalnis S (2005) Techniques for efficient road-network-based tracking of moving objects. IEEE Trans Knowl Data Eng 17(5):698–712
34. Chawathe SS (2007) Segment-based map matching. In: 2007 IEEE Intelligent vehicles symposium, IEEE, pp 1190–1197
35. Alt H, Efrat A, Rote G, Wenk C (2003) Matching planar maps. J Algorithms 49(2):262–283
36. Brakatsoulas S, Pfoser D, Salas R, Wenk C (2005) On map-matching vehicle tracking data. In: Proceedings of the 31st international conference on very large data bases, pp 853–864
37. Lou Y, Zhang C, Zheng Y, Xie X, Wang W, Huang Y (2009) Map-matching for low-sampling-rate gps trajectories. In: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems, pp 352–361
38. Bruna J, Zaremba W, Szlam A, LeCun Y (2013) Spectral networks and locally connected networks on graphs. arXiv:1312.6203
39. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. arXiv:1606.09375
40. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv:1609.02907
41. Atwood J, Towsley D (2015) Diffusion-convolutional neural networks. arXiv:1511.02136
42. Niepert M, Ahmed M, Kutzkov K (2016) Learning convolutional neural networks for graphs. In: International conference on machine learning, PMLR, pp 2014–2023
43. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. arXiv:1710.10903
44. Zhou Z-H, Feng J (2017) Deep forest. arXiv:1702.08835
45. Duan Y, Yisheng L, Wang F-Y (2016) Travel time prediction with lstm neural network. In: 2016 IEEE 19th international conference on intelligent transportation systems (ITSC), IEEE, pp 1053–1058

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Shu Lin (M'07) received the MSc degree in control science and engineering from Shandong University, Jinan, China, in 2006, and the PhD degree at the Delft Center for Systems and Control of Delft University of Technology, Delft, The Netherlands, in 2011. From 2011 to 2013, she was a postdoc researcher in the Department of Automation of Shanghai Jiao Tong University in Shanghai, China. Currently, she is an associate professor at School of Computer Science and Technology in University of Chinese Academy of Sciences in Beijing, China. Her research interests mainly include modeling, optimization, and control for intelligent traffic networks.

Yanyan Xu received the Ph.D. degree from the Department of Automation, Shanghai Jiao Tong University in 2015, and the M.S. and B.E. degrees in Shandong University in 2010 and 2007, respectively. He is now an associate professor in the AI Institute, Shanghai Jiao Tong University. Prior to joining SJTU, he was a postdoctoral associate in the Human Mobility and Networks (HuMNet) Lab in the Department of Urban and Regional Planning at UC Berkeley and the Department of Civil and Environmental Engineering at MIT from 2015 to 2020. He was also a guest postdoctoral fellow in the Energy Analysis and Environmental Impacts Division, Lawrence Berkeley National Laboratory from 2017 to 2018. His research lies in human mobility and urban computing, with particular emphasis placed on the use of massive trajectory data in Smart City, Transportation, Energy, and Environment, from an interdisciplinary perspective. His work has been published in Nature Energy, Science Advances, J. Roy. Soc. Interface, IEEE Trans. ITS, TRC, and IJCAI, among others.

Shengjian Zhao received the M.S. degree in software engineering from University of Chinese Academy of Sciences, Beijing, China, in 2021. He is currently working at Alibaba as an Algorithm Engineer. His research interests include time series prediction, abnormal detection, and machine learning.

Yibing Wang received the B.Sc. degree in electronics and computer engineering from Sichuan University, Chengdu, China; the M.Eng. degree in automatic control engineering from Chongqing University, Chongqing, China; and the Ph.D. degree in control theory and applications from Tsinghua University, Beijing, China. He was a Postdoc/Research Fellow/Senior Research Fellow with the Dynamic Systems and Simulation Laboratory, Department of Production Engineering and Management, Technical University of Crete, Greece, from 1999 to 2007. He was a Senior Lecturer at the Department of Civil Engineering, Monash University, Australia, from 2007 to 2013. Since 2013, he has been a Full Professor with the Institute of Intelligent Transportation Systems, Zhejiang University, China. His research interests include road traffic flow modeling, traffic surveillance, urban and freeway traffic control, connected & automated vehicles (CAV). He serves as a Senior Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and as an Associate Editor for Transportation Research Part C: Emerging Technologies. His research has been supported by the European Commission, the Discovery program of the Australian Research Council, the National Natural Science Foundation of China, and the National Key R&D Program of China.

Jungang Xu is a full professor in School of Computer Science and Technology, University of Chinese Academy of Sciences. He received his Ph.D. degree in Computer Applied Technology from Graduate University of Chinese Academy of Sciences in 2003. His current research interests are automated machine learning, computer vision and intelligent transportation systems.

Authors and Affiliations

Shu Lin[1] · Yanyan Xu[2] · Shengjian Zhao[1] · Yibing Wang[3] · Jungang Xu[1]

Yibing Wang
wangyibing@zju.edu.cn

Jungang Xu
xujg@ucas.ac.cn

[1] School of Computer Science Technology, University of Chinese Academy of Sciences, Beijing 101408, China

[2] The MoE Key Laboratory of Artificial Intelligence in AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China

[3] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China