ANNALS OF APPLIED SCIENCES
MIDITERRANEAN JOURNALS

Original Research

DOI: [10.55085/aas.2023.712](https://doi.org/10.55085/aas.2023.712)

A Deep Spatiotemporal Model for Travel Time Prediction

Songyu Han[1], Zhiqiang Lv[2, 3], Liping Fu[2]
[1] Mellon College of Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2] College of Computer Science & Technology, Qingdao University, Qingdao 266071, China
[3] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

ABSTRACT

Accurate and reliable travel time prediction is important for promoting the development of urban public transportation, ensuring public travel safety, and establishing smart cities. In travel time prediction, studying spatiotemporal correlation features can help us better understand the dynamic changes and spatial dependencies in traffic data, and it can explain the patterns and trends in vehicle travel. Studying the external factors that influence vehicle travel can help us comprehensively consider the complexity of the transportation system, and incorporating these factors into prediction models can enhance the accuracy and robustness of the models. Therefore, this article proposes a novel deep spatiotemporal model for travel time prediction called DeepSTM-TTP. The architecture of this model consists of three parts: spatial-temporal convolution mechanism, external factor mechanism, and multitask learning mechanism. The spatiotemporal convolution mechanism is used to capture the spatiotemporal correlation of the trajectory external factor mechanism is used to handle the external information in the trajectory. A multitask learning mechanism achieves a balance between local path travel time prediction and whole path travel time prediction. The model fully considers the spatial-temporal correlation of the original GPS location sequence with external information. The experimental results on real datasets demonstrate that the model proposed in this article outperforms four well-known travel time prediction models, including a statistical model (HA), a machine learning model (GBDT), and two deep learning models (DeepTTE and DeepTTE-RNN).

KEYWORDS: Travel time prediction; Deep Spatial-Temporal Model; Multitask learning; Deep learning.

Received: 28 Apr 2023
Revised: 07 July 2023
Accepted: 13 July 2023
Published: 10 Aug 2023

Correspondence:
Zhiqiang lv. College of Computer Science & Technology, Qingdao University, Qingdao 266071, China
Email: 2019025929@qdu.edu.cn

Cite this article as: Han S, Lv Z, Fu L. A Deep Spatiotemporal Model for Travel Time Prediction. Ann Appl Sci. 2023;2:712. https://doi.org/10.55085/aas.2023.712

Authors' contributions
The participation of each author corresponds to the criteria of authorship and contributorship emphasized in the Recommendations for the Conduct, Reporting, Editing, and Publication of Scholarly work in Medical Journals of the International Committee of Medical Journal Editors. Indeed, all the authors have actively participated in the redaction, the revision of the manuscript, and provided approval for this final revised version.

Acknowledgments
None

Funding
No funding was received from any organization to conduct the present study.

Conflict of interest
The author declares that there is no conflict of interest regarding the publication of this article.

1. INTRODUCTION

The traffic demand [1] is increasing with the rapid development of the urban economy and the rapid increase in urban population, leading to traffic congestion problems [2]. As shown in Figure 1, the cities (Chongqing, Guiyang, and Beijing) have become the three most congested cities in China according to the 2020 China Urban Traffic Report jointly released by Baidu Maps and various departments. Traffic congestion easily leads to accidents such as rear-end collisions, significantly affecting the operation efficiency of vehicles and normal travel [3]. At the same time, the emission of exhaust gas has increased dramatically with traffic congestion, which causes a series of environmental pollution, resource waste, and other problems, seriously affecting city development.

At present, China's urban travel modes mainly include taxis, buses, subways, and shared bicycles, but the most popular means of transportation are still taxis or buses [4]. Taxis are easy to use and less affected by geographical restrictions and save time spent parking. Buses can be used in a wide range of applications and basically cover the entire urban area [5]. Compared with the subway, their investment is small and easy to implement. And people can ride at a low price, which is generally acceptable and has a wide audience. These two means of transport account for the central part of public transportation and play an essential role in urban transport planning [6]. However, people will wait for a long time when taking a taxi, and they cannot know the exact arrival time of the vehicle. These situations will increase people's anxiety. People will be late for work or even miss some important meetings, which will reduce people's confidence in urban public transport. It is not conducive to the development of urban public transport. Therefore, real-time and accurate taxi travel time prediction is particularly important.

When people want to take a taxi, they can find the time through the time prediction system [7]. They can plan their trip in time, alleviating anxiety during the waiting period. In addition, drivers can choose the segments with shorter times according to the travel time of different roads. This can avoid traffic congestion and alleviate traffic pressure [8]. In addition, accurate and real-time travel time prediction can help the traffic department timely obtain the traffic status in a short period of time. Thus, it can reasonably arrange department personnel in advance and better manage traffic emergencies. This can also help prevent a wide range of traffic congestion and traffic accidents, improve people's travel efficiency, and ensure people's travel safety [9]. With the reduction of traffic congestion, the utilization efficiency of resources can be improved to a certain extent; energy waste can be avoided [10]. The emission of pollutants in vehicle exhaust can be reduced, which is conducive to improving urban air quality. Therefore, accurate travel time prediction has certain practical significance.

In the research field of vehicle travel time prediction, this article fully considers the spatiotemporal correlation features of vehicle travel data and the external factors influencing vehicle travel. As a result, a high-accuracy spatiotemporal prediction model is constructed. Furthermore, this article introduces a multitask learning module to capture the feature relationships between experienced travel time and instantaneous travel time, further improving the accuracy of the prediction model [11]. The contributions of this article can be summarized as follows:

- We propose a Deep Spatial-Temporal Model for Travel Time Prediction (DeepSTM-TTP) based on a deep spatial-temporal neural network, considering the waiting time and travel time. The model mainly comprises three parts: an external factor module, a spatial-temporal convolution module, and multitask learning [12] module.
- We clean and process GPS point data and catch the passenger carrying status. The point data is matched to a driving trajectory of the vehicle from the no-load status to the full-load status.
- We propose using the combination of experienced and instantaneous travel time to avoid the impact of error accumulation and data sparsity.
- We conduct extensive experiments on public traffic network datasets to show that the proposed method outperforms state-of-the-art methods.

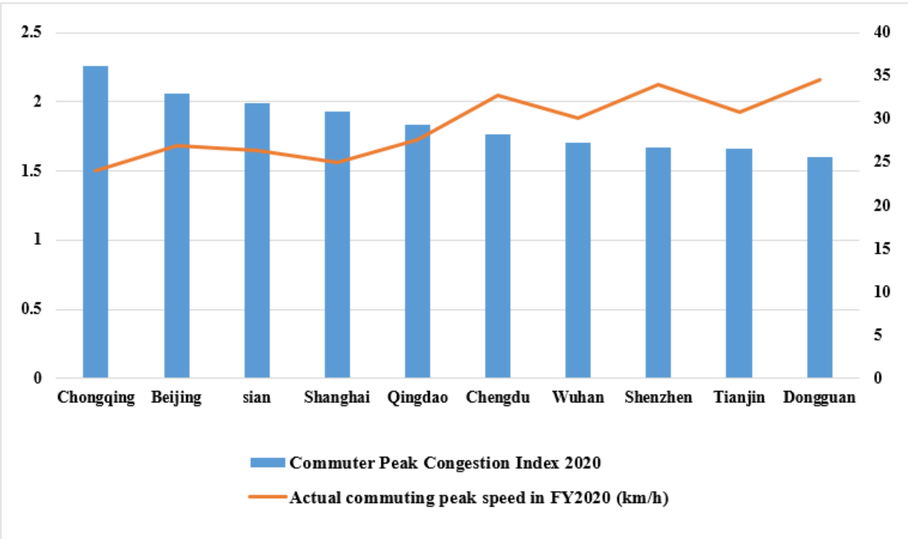


Figure 1. Traffic Congestion in China's Major Cities in 2020.

1.1. Related work

1.1.1. Travel time prediction based on vehicle trajectory

In the prediction method based on vehicle trajectory, people install loop detectors at both ends of experienced travel time and use the loop detectors to study travel time. When a vehicle passes on the road, the loop detectors record the passing time and estimate the vehicle's speed by calculating the time difference, respectively. Jia et al. [13] proposed a PeMS algorithm, which uses the vehicle speed obtained by the loop detector to estimate travel time. Asif et al. [14] used vector machines in combination with historical data to predict the passing time of roads. Gao et al. [15] combined a support vector machine (SVM) with a genetic immune algorithm and used SVM to establish a prediction model. This model used the genetic immune algorithm to optimize parameters in order to prevent overfitting. Because few roads have loop detectors installed in cities, this method cannot be used to predict the travel time of the whole city. Many studies use floating car data. In these studies, GPS trajectory is used to predict travel time. However, there are obvious shortcomings in the travel time prediction method based on vehicle trajectory data. It does not consider the connection between paths, each intersection, or the impact of traffic lights on travel time, so its accuracy is not high. Jenelius et al. [16] used the maximum likelihood estimation method to predict the travel time of obtained low sample trajectory data, considering the traffic patterns between paths. However, the prediction method based on experienced travel time is used to predict the travel time of a single experienced travel time, and the total time is obtained by adding and summing the time of a single experienced travel time. In real life, the traffic at the intersection has a certain complexity, which is very important for a path. The prediction method based on experienced travel time has a large error.

1.1.2. Travel time prediction based on experienced travel time and instantaneous travel time

Given the shortcomings of methods based on vehicle trajectory, researchers proposed a travel time prediction method based on experienced travel time and instantaneous travel time. Xiao et al. [17] explored the differences between travel time estimation based on detector data and automatic vehicle identification (AVI) data and compared instantaneous travel time and experienced travel time. The study revealed that, compared to uncongested conditions, the difference in instantaneous travel time and experienced travel time during congested conditions ranged from 6% to 17%. Zhang et al. [18] proposed a simple and efficient travel time prediction method based on probe vehicle data, which matches large-scale spatiotemporal traffic patterns for predicting instantaneous and experienced travel times. Osman et al. [19] compared the performance of two deep learning models (Multi-Layer Perceptron Neural Networks (MLP-NN) and Long Short-Term Memory Networks (LSTMN)) in predicting both instantaneous and experienced travel times for buses and found that the uncertainty in traffic conditions significantly impacts the predictions. The study of Lin et al. [20] focused on travel time prediction for signalized corridors, demonstrating that a model combining exponential smoothing, artificial neural network (ANN) techniques, and Bayes algorithms effectively captures both instantaneous and experienced travel times. These research findings highlight the importance of considering both instantaneous and experienced travel times in travel time prediction models, as they are closely related to traffic conditions and have strong mutual influences. Therefore, this study takes into account the spatiotemporal correlation features of traffic data, the relationships with external influencing factors [21], and the impacts of both instantaneous and experienced travel times on the entire modeling process when developing travel time prediction models.

1.1.3. Travel time prediction based on deep learning

In recent years, with the development of artificial intelligence, deep learning technology has appeared in various studies, such as natural language processing, computer vision, and speech recognition. Some scholars apply the method of deep learning to travel time prediction [22]. Wang et al. [23] proposed a deep learning framework for travel time estimation (DeepTTE), which uses convolutional neural networks to obtain the spatial characteristics of historical GPS trajectories and Long Short-Term Memory (LSTM) to extract the time characteristics. Qiu et al. [24] proposed a Gated Recurrent Unit (GRU) method to predict travel time and use the velocity characteristics to represent the feature information of adjacent road segments. Zhang et al. [25] proposed a new assistant monitoring model based on deep learning technology. By introducing a double-interval loss function, the time marker information in the trajectory data can be fully utilized, and different feature information can be extracted automatically and effectively to predict the travel time accurately. Lan et al. [26] proposed an end-to-end multitask deep neural network model (Deep Image to Time, DeepI2T), which mainly uses trajectory data and geomorphic layout images to learn the travel time of a route. This method does not use map matching and road network but combines the image layout in the grid with the direction of the vehicle to achieve the purpose of the travel time prediction. Zou et al. [27] proposed a travel time prediction method based on multimodal integration of large urban data. The method first extracts the eigenvectors from the multimodal data as input to the model and then uses the gradient-enhanced decision tree model and the deep neural network model to process the low-dimensional and high-dimensional features. Finally, it integrates the two models based on the integration method. Traffic signals also have a significant impact on travel times [28]. Generally speaking, the change in traffic signal is one of the important causes of traffic congestion. When traffic congestion occurs, taxi travel times are extended. The more traffic signals a taxi encounters, the longer the trip will take. Tang et al. [29] proposed a tensor-based spatiotemporal model for citywide travel time estimation using large and sparse GPS trajectories received from cabs. By reconstructing the tensor, it is possible to know not only the travel times under different traffic conditions but also the probability of occurrence of the corresponding traffic conditions.

2. METHODOLOGY

2.1. Problem definition

This article first cleans the original GPS data, deletes the error points, and then matches the trajectory. A trajectory is a collection of consecutive GPS points (each containing longitude, latitude, and timestamp): $Q = \{q_1, q_2, \ldots, q_n\}$. Secondly, each trajectory also contains the start time, license number, date, whether or not to carry passengers, and so on. We also propose an important attribute: travel distance. $dis(q_1, q_2)$ is used to represent the total distance of a driving trajectory as shown in equation (1), where $dis(q_i, q_{i+1})$ represents the straight line distance between q_i and q_{i+1}.:

$$dis(q_1, q_n) = \sum_{i=1}^{n-1} dis(q_i, q_{i+1}). \qquad (1)$$

According to the above definition, the problem of taxi travel time prediction is described as follows: Set T as the traveling trajectory on the travel segment, given P as the query path, and path T is related to path P. Our goal is to learn the model parameters using a given historical route training model to estimate the travel time of the entire route P. This process can be represented by equation (2), where f represents the mapping relationship represented by the model parameters and TravelTime represents the travel time of the taxis:

$$[T, P] \xrightarrow{f} TravelTime. \qquad (2)$$

2.2. Model design

As shown in Figure 2, this article proposes a taxi travel time prediction model DeepSTM-TTP based on combining a traditional convolutional neural network and a temporal convolution network. The model consists of three parts: spatial-temporal convolution mechanism, external factor mechanism, and multitask learning mechanism. The model fully considers the spatial-temporal correlation of the original GPS position sequence and external information (such as license plate number and date). The external factor mechanism is used to process the external information in the trajectory. The spatial-temporal convolution mechanism is used to capture the spatial-temporal correlation of trajectories. The output of the external factor mechanism and the spatial-temporal convolution mechanism is taken as the input of the multitask learning mechanism. Finally, the multitask learning mechanism is used to achieve the balance between the local path travel time prediction and the experienced travel time prediction, and the final prediction time is obtained.

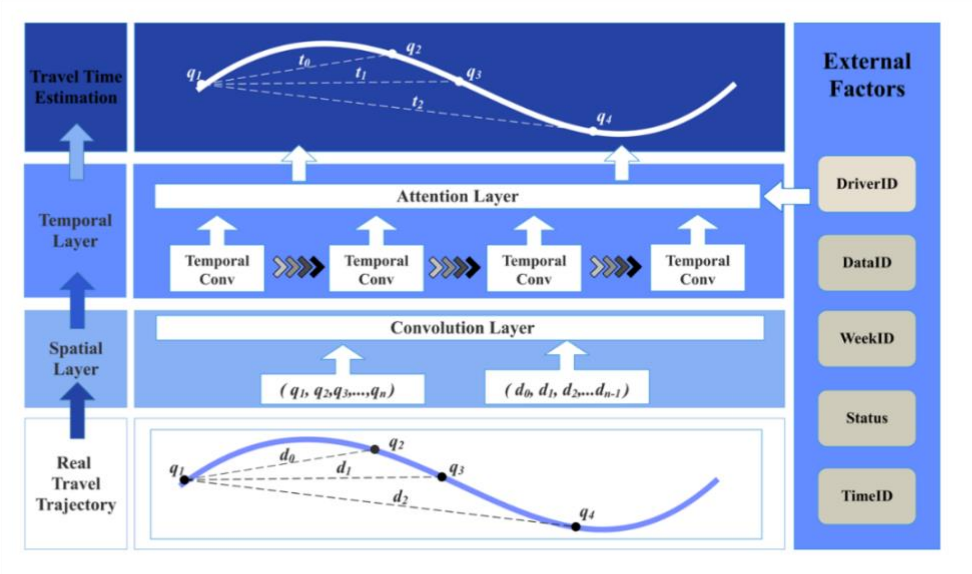


Figure 2. DeepSTM-TTP framework. Convolutional kernel size is 3*3, TCN convolutional kernel size is 3*3, expansion factor is $2^n(n = 0,1,2,\ldots)$, number of nodes is 307, and the feature dimension is 1.

2.2.1. External factor mechanism

Different drivers have different driving habits. The smoothness of roads varies at different time stages. Passengers who are in a hurry to travel may urge drivers to speed up. Therefore, it is necessary to consider the influence of external factors, such as license plate number, date, cab passenger status, and departure time, when predicting the travel time of road vehicles. This article proposes an external factor mechanism to fully catch these external factors. Because these external factor values are classified values, they cannot be directly input into the neural network. We use the embedding method to convert these classification variables $l \in [L]$ into embedded space RM×1. This way, the input dimension can be effectively reduced and the training speed can be improved using the embedding method. Finally, the embedded vector is connected with the travel distance $dis(q_1, q_2)$ as the output ext of the external factor mechanism.

2.2.2. Spatial-temporal convolution mechanism

The spatial-temporal convolution mechanism is introduced. The model uses the convolution neural network layer and temporal convolution network layer to capture the spatial dependence and temporal correlation between trajectory data, respectively, in order to achieve accurate prediction and analysis of temporal series data.

The historical data used in this article is a collection of GPS points, $Q = \{q_1, q_2, \ldots, q_n\}$, each q_i containing longitude, latitude, and c information. In the research, it is necessary to extract the spatial characteristics of the original GPS trajectory data. Previous studies proposed the use of graph embedding to extract spatial information of adjacent regions. In the model, we use a nonlinear function combined with a convolutional neural network (CNN) to obtain spatial features. In CNN, several different feature maps can be obtained by convolution operation through different convolution filters, and neurons in the same feature map share weight. The advantage of sharing weights is to reduce the occurrence of overfitting and layer-to-layer connections in the network. Compared with other deep network models, traditional CNN models are better used in face recognition, target tracking, natural language processing, speech recognition, and so on.

In the model, a nonlinear function is used to map the path location information to the vector $ploc_i \in 16$:

$ploc_i = \tanh\,(W_{ploc} < q_{i,lat}, q_{i,lon} >)$, (3)

Where < > indicates the connection operation of the ith path, W_{ploc} indicates the learning parameter matrix, and tanh is the activation function. The output vector $ploc_i$ is used as the input of the CNN. The traditional CNN model used in this article includes two convolution layers, two pooling layers, and one upsampling layer. The convolution layer fully extracts the spatial information features of the path and then compresses the features through the pooling layer (maximum pooling method is used in the experiment) to extract the main features. The combination of the convolution layer and pooling layer effectively reduces the complexity of parameters and optimizes the model. Finally, the dimension of the feature map is enlarged by using the upper sampling layer to make it have a higher resolution. The output of the convolutional network layer is $conv_2$.

$$conv_i^1 = \sigma(W_{conv}^{(1)} \cdot ploc + b^{(1)}), \quad (4)$$

$$conv_i^2 = \sigma\left(W_{conv}^{(2)} \cdot conv_i^1 + b^{(2)}\right), \quad (5)$$

where W_{conv} and b are two parameters and σ is an activation function. The final characteristic diagram is obtained by convoluting the network layer and recorded as spa.

In order to further deal with the time correlation between the segments, we use the temporal convolution network (TCN). The input of the TCN is the characteristic diagram spa and the embedded vector ext. The update state is shown in the following formula:

$$r_i = \sigma(W_s \cdot spa + W_e \cdot ext + W_h \cdot r_{i-1}), \quad (6)$$

where W_s, W_e, W_h are the learning parameters used in the TCN layer and σ is an activation function. Finally, the spatial-temporal characteristic sequence {r1, r2, r3, ..., r|T|+k+1} is obtained. Compared with LSTM and gated loop unit, the TCN has higher accuracy in time prediction of sequence model, and its structure is simpler and clearer.

2.2.3. Multitask learning mechanism

We take the multitask learning mechanism part as the third part of the model and combine the feature sequence with the relevant $\{r_1, r_2, r_3, \ldots, r_{|T|+k+1}\}$ external factors. Because the instantaneous travel time prediction method ignores the influence of intersections and signal lights, it simply sums the travel time of each segment. In real life, when we drive, the time for waiting for the red light and crossing the intersection account for a part of the total travel time, which has a certain impact on travel time prediction. Therefore, ignoring the time of vehicles passing the intersection and waiting for the red light makes the results inaccurate. In addition, the problem of data sparsity will occur in the entire segment time prediction method [30]. Therefore, in the part of the multitask learning mechanism, we predict the travel time of instantaneous travel time and experienced travel time, respectively. In the instantaneous travel time prediction, we use two fully connected network layers to map the spatial-temporal feature sequence $\{r_1, r_2, r_3, \ldots, r_{|T|+k+1}\}$ to the temporal sequence $\{h_1, h_2, h_3, \ldots, h_{|T|+k+1}\}$, h_i representing the travel time prediction of the instantaneous travel time $q_i \to q_{i+1} \to \cdots \to q_{i+k-1}$.

If there are traffic lights, intersections, or other complex situations on a certain road segment, there may be traffic jams and other situations on the road segment and the driving time on the road segment may increase while the travel time on the road segment with simple situations will be shorter [31]. Therefore, we should pay more attention to complex roads. In the part of experienced travel time prediction, we add an attention mechanism [32]. The output $\{r_1, r_2, r_3, \ldots, r_{|T|+k+1}\}$ of the spatial-temporal convolution mechanism is taken as the input. Input into the attention mechanism to set different weights according to the importance of different paths to the entire path and obtain the vector r_{ext}. The attention mechanism is essentially a weighted sum operation of the sequence $\{r_1, r_2, r_3, \ldots, r_{|T|+k+1}\}$, as shown in equations (7), (8), and (9):

$$r_{att} = \sum_{i=1}^{|T|-k+1} \alpha_i \cdot r_i, \quad (7)$$

$$\alpha_i = \frac{e^{z_i}}{\sum_j e^{z_j}}, \quad (8)$$

$$z_i = < \sigma_{ext}(ext), r_i >. \quad (9)$$

The attention mechanism is widely used, including image processing, speech recognition, natural language processing, and other fields. Finally, we input r_{ext} into the full connection layer network to obtain the travel time h_{en} of the experienced travel time.

This article uses an end-to-end approach to train our model. In order to obtain the optimal training effect, in the training phase, we predict the travel time of all instantaneous travel time and the experienced travel time at the same time and define two loss functions. The first loss function is defined as the average of the relevant loss functions of all instantaneous travel time.

$$L_l = \frac{1}{|T|-k+1} \sum_{i=1}^{|T|-k+1} \left| \frac{h_i - (q_{i+k-1} \cdot t_s - q_i \cdot t_s)}{q_{i+k-1} \cdot t_s - q_i \cdot t_s + \omega} \right|. \quad (10)$$

The second loss function is defined as the relevant loss function of the experienced travel time.

$$L_e = \frac{|h_{en} - (q_{|t|} \cdot t_s - q_1 \cdot t_s)|}{q_{|t|} \cdot t_s - q_1 \cdot t_s}. \quad (11)$$

In the training phase, the loss function is defined as the weighted sum of L_l and L_e. The loss is minimized through the training model. α represents a coefficient to balance the weights of L_l and L_e.

$$loss = \alpha \cdot L_l + (1 - \alpha) \cdot L_e. \quad (12)$$

2.3. Datasets and experiment setting

The data used in the experiment is GPS trajectory data generated by over 14,000 taxis in Chengdu, China, in July 2014. The original dataset was collected from GPS devices installed in vehicles operating on the road network. These devices typically collect one record at a fixed time interval, forming the GPS trajectory data. However, the original data had some issues, such as data noise, outliers, and data duplication. To ensure the quality and accuracy of the data, data cleaning was performed in this study. Through the data cleaning steps, we obtained a reliable dataset suitable for further analysis and modeling. The data cleaning process included the following steps:

Trajectory segmentation and deletion: We removed the too-short trajectories. We segmented excessively long trajectories, ensuring that the longest trajectory distance in the dataset was 20 kilometers and the shortest trajectory distance was 2 kilometers.
Data resampling: We resampled the trajectory data to ensure that the distance gap between two consecutive points was around 200 to 400 meters.
Outlier detection and duplication removal: We identified and removed outliers by examining the data for possible errors or abnormal data points. Additionally, we eliminated duplicate records and duplicate data points in the dataset.
One piece of data contains 11 pieces of field information, forming a driving trajectory. Each trajectory includes latitude, longitude, timestamp, date, trajectory start time, license number, passenger status, and other information. There are about 500,000 trajectories of data every day. In the experiment, the training set uses data from the first 18 days, the evaluation set uses data from the middle five days, and the test set uses data from the last five days. The interpretation of each field in trajectory data is shown in Table 1.

Table 1. Introduction to Chengdu Dataset.

Subsegment	***Meaning***
time_gap	The time interval between each GPS sampling point and the first GPS sampling point in a trajectory path, in seconds
dist	The total length of the trajectory path in kilometers
Lats, lngs	The longitude and latitude information of each GPS sampling point in the path
driverID	License plate number information
dataID, weekID	Sampling date
Status	Vehicle passenger carrying status, 0.0 represents the taxi no-load status, 1.0 represents the taxi passenger carrying status
timeID	Vehicle trajectory start time

The computer configuration used in the experiment is as follows. The CPU we used is Intel (R) Xeon (R) CPU E5–2620 v4 @ 2.10GHz,32-core. The GPU we used is NVIDIA Corporation GV100GL [Tesla V100 DGXS 32GB] × 8. The total memory of the computer is 128G. The experimental result is the average of 20 training processes, and each training process has experienced 150 epochs. The batch size is 64. The initial learning rate for training is 0.001. Adam is selected as the gradient optimizer for the training process. We chose the commonly used loss function L2 for this experimental training process.

2.4. Experimental evaluation index

In the analysis of experimental prediction results, appropriate indicators are needed to evaluate the performance of the model. Assuming that the road travel time predicted by the model is $\hat{y}_i$ and the actual road travel time is y_i, the article uses the following three evaluation indexes to evaluate the prediction performance of the model.
(1) Mean Absolute Error (MAE) is the mean of the deviation between the predicted value and true value, which measures the error between the predicted value and true value. The smaller the MAE, the better the prediction model. MAE is defined in the following equation:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \quad (13)$$

(2) Root Mean Square Error (RMSE) is the square root of the ratio of the squared deviation of the predicted value to the true value and the number of times predicted, which is used to measure the deviation between the predicted value and true value. The smaller the RMSE, the smaller the error between the predicted value and the true value. The model is defined in the following equation:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \quad (14)$$

(3) Mean Absolute Percentage Error (MAPE) is the average of the ratio of the predicted value to the true value and the deviation from the true value. It measures the error between predicted value and true value and solves the robustness problem in the evaluation index effectively. It is defined as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|}. \quad (15)$$

(4) The symmetric mean absolute percentage error (SMAPE) is the average of the ratio of predicted value to true value and the deviation from the sum of true value and predicted value. It is a correction indicator for the problem of MAPE, which can better avoid the problem that MAPE is calculated too big because the real value is small. It is defined as follows:

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}. \quad (16)$$

3. RESULTS

3.1. Benchmarking method

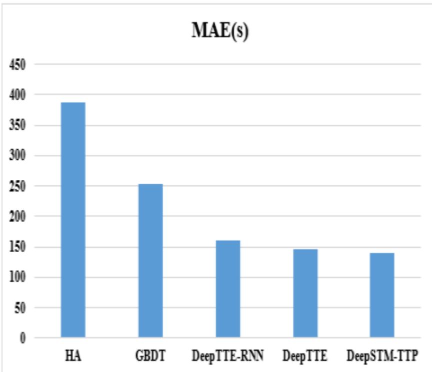
This article selected four models for comparison with DeepSTM-TTP, including one statistical model (HA), one machine learning model (GBDT), and two deep learning models (DeepTTE and DeepTTE-RNN). The selection of these models is based on the fact that each model represents different methods and techniques with their own advantages and characteristics. The HA serves as a simple benchmark model that uses historical average values for prediction. The GBDT has strong generalization and learning capabilities, allowing it to automatically learn features and patterns from the data. The DeepTTE and DeepTTE-RNN excel in handling complex sequential data, enabling them to automatically extract temporal and spatial features and complex relationships within the data. By comparing these models, the performance of each model and its comparison with DeepSTM-TTP in travel time prediction tasks can be evaluated, providing insights into result validation and further improvements and optimizations. The model is described and designed as follows:

(1) Prediction model based on historical average data (HA) [33]: We derive historical average speeds from historical trajectories, select 2:00 to 4:00 p.m. every Tuesday as the test time, and predict taxi travel time based on the start time and historical speed of the vehicle.

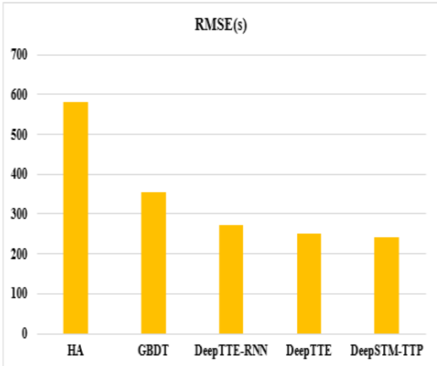
(2) Gradient-Enhanced Decision Tree Model (GBDT) [34]: GBDT is a widely used integrated method for road travel time prediction. Since it only handles equal-length sequences, and the original GPS sequence we obtained is variable in length. We sampled each sequence into 128 fixed-length sequences and used them as input to the model along with other external factors.

(3) DeepTTE [23]: It is a good prediction model for road travel time prediction. The original GPS trajectory information is mapped nonlinearly. And then, CNN and LSTM are used to extract the spatial-temporal characteristics of each road segment.

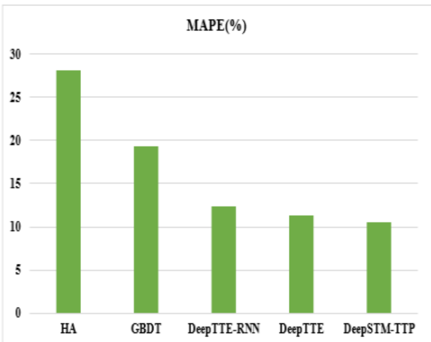
(4) DeepTTE-RNN: It is a modification of DeepTTE with the same input as DeepTTE. We change LSTM to RNN to study their performance in extracting time characteristics.



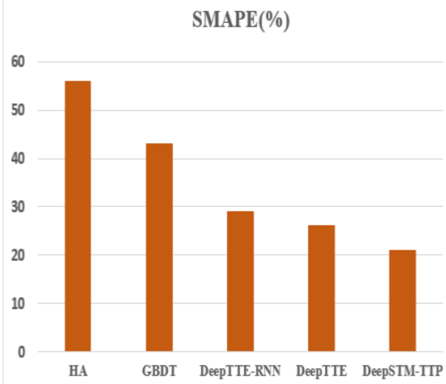
(a) Comparison of MAE results



(b) Comparison of RMSE results



(c) Comparison of MAPE results



(d) Comparison of SMAPE results

Figure 3. Comparison of experimental results. The x-axis represents the models, while the y-axis represents the values of evaluation indicators. The four indicators are regression indicators, and their values indicate the degree of error between the predicted values of the model and the true values. Therefore, smaller values of the four indicators indicate better prediction performance of the model.

Experiments have been performed to evaluate the performance of DeepSTM-TTP on real trajectory datasets. The system running in this chapter is a Linux system. The programming language is Python and the library version is pytorch1.2. In the training phase, this article uses MAE, RMSE, MAPE, and SMAPE to evaluate the prediction results. We use the Adam optimization algorithm to train the model with five times cross-validation. During the training of the model, we set the learning rate to 0.001, the batch size to 64, and the training epoch is 200. The experimental results are shown in Figure 3. When using the HA method for prediction, the error is large and the accuracy is relatively low. As a widely used integration method, GBDT has a significantly higher prediction accuracy than HA. But its temporal dependence is ignored when processing data, resulting in poor experimental results. Based on the experimental results, an end-to-end method is used for prediction with high accuracy. DeepTTE and DeepTTE-RNN take into account the spatial-temporal correlation of trajectory data and the influence of external factors. In the experiment, DeepTTE and DeepTTE-RNN use LSTM and RNN to extract time characteristics, respectively. MAPE is 11.26% and 12.37%, reflecting that LSTM performs better than RNN in processing time series data. Finally, our model DeepSTM-TTP performs better than DeepTTE with an error of only 10.58%. A large number of simulation experiments have been carried out to further study the impact of each part of the model on the prediction results. We performed ablation experiments on the trained deep learning model, DeepSTM-TTP, by removing certain external factors from the input. This allows us to determine which attributes impact the experimental results most and how their contributions affect the final outcome. Ablation experiments can help us better understand how different external attributes affect the final outcome. We used weather and velocity attributes as the control group and performed multiple experiments using weather-only, velocity-only, and weather and velocity attributes as the experimental groups while recording the results. In the external factor mechanism section, to better capture the feature information of adjacent roads, we added the velocity variable to assist in extracting the feature information of nearby roads [35] and studied the impact of weather and velocity characteristics on time prediction, respectively, as shown in Figure 4. In the DeepSTM-TTP model, errors decrease by 0.56% and 1.32% by adding weather and velocity attributes. When both attributes are added at the same time, the error is reduced by 1.79%, showing the validity of both attributes. This aligns with the reality that in rainy or foggy weather, drivers slow down and travel times are relatively longer. And when adjacent segments encounter traffic congestion or traffic accidents, the adjacent segments will certainly be affected. Therefore, when the characteristic information of adjacent road segments is added through velocity features, the spatial characteristics of adjacent road segments are incorporated into the time prediction of the current path, which greatly improves the prediction accuracy.

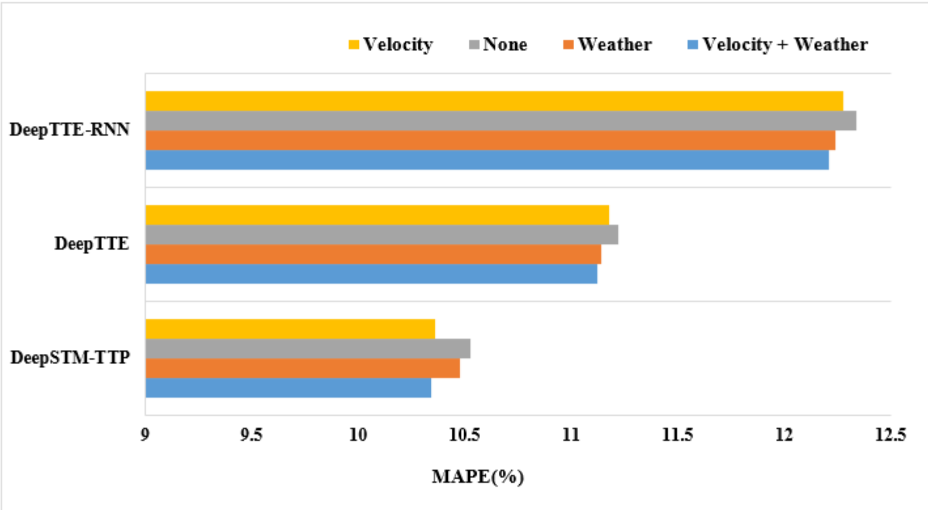


Figure 4. Changes in MAPE Indicators when adding weather and velocity attributes.

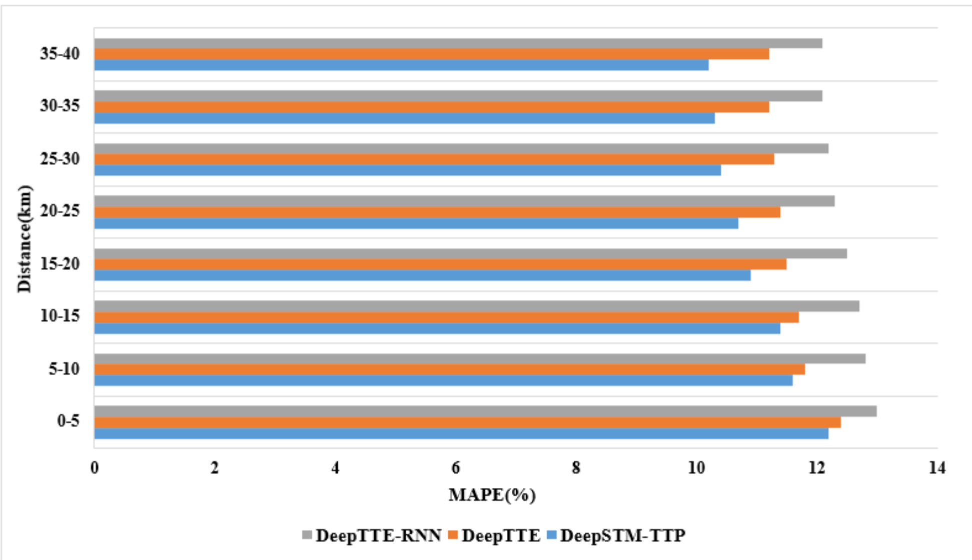


Figure 5. Changes of MAPE Indicators with distance.

In addition, we studied the relationship between trajectory path length and prediction accuracy, as shown in Figure 5, and compared DeepTTE and DeepTTE-RNN with our model. As the length of the path increases, the feature information obtained during prediction increases and learning ability increases, so the MAPE of the model decreases. DeepSTM-TTP has no obvious advantage over other models when the path length of the process is short. When the path length of the process is longer than 20 kilometers, the prediction error is low, which reflects the better prediction performance of our model in medium-long distance paths.

4. CONCLUSIONS

In this article, we first introduce the prediction of taxi travel time on urban roads. According to the actual situation of passengers taking taxis, the predicted time includes the waiting time and travel time. Through cleaning the acquired taxi GPS point data, the wrong points are deleted. And the trajectory is processed from no load to full load according to the taxi passenger carrying status. This article proposes the DeepSTM-TTP model to deal with the temporal correlation, spatial dependence, and other external attributes between paths. In the model, firstly, nonlinear functions are used to map the trajectory path to the grid, and the spatial features in the path are extracted through CNN. Then, TCN is used to extract the time features between the paths. In the prediction phase, we input the spatial-temporal feature sequence obtained into the stacked fully connected network and predict the travel time by combining the instantaneous travel time with the experienced travel time. Finally, a large number of simulation experiments are carried out. Through the analysis of the experimental results, it is verified that our spatial-temporal model has excellent prediction performance when dealing with the problem of time prediction. However, our work has some limitations. Firstly, many factors affect taxi travel time, such as the number of traffic lights on the driving route, the number of turns, the congestion coefficient of the road, and the weather. Our work only considers the license plate number, date, passenger status of the taxi, and departure time of the trip. Secondly, further improvement is needed in the feature extraction methods for external factors to enhance the accuracy of the model. Based on the above limitations, we can improve prediction accuracy through the following methods. First, we can collect and process more external factor data in the external factor mechanism. Second, for more external factors, we can use an additional convolution module to perform preliminary feature extraction and then combine the subsequent attention mechanism to fully model the external factors. Third, more convolution layers will improve the prediction accuracy but will bring the problem of gradient disappearance or gradient explosion. We can stack basic spatial-temporal convolution modules in a residual linking to improve the final prediction accuracy.

REFERENCES

[1] Lee JM, Kim JD. A Generative Model for Traffic Demand with Heterogeneous and Spatiotemporal Characteristics in Massive Wi-Fi Systems. Electronics. 2022 Jun 10;11(12):1848. doi: 10.3390/electronics11112848

[2] Jilani U, Asif M, Rashid M, Siddique AA, Talha SMU, Aamir M. Traffic congestion classification using GAN-Based synthetic data augmentation and a novel 5-Layer convolutional neural network model. Electronics. 2022 Jul 22;11(15):2290. doi: 10.3390/electronics11152290

[3] Xu Z, Lv Z, Li J, Sun H, Sheng Z. A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. IEEE Intelligent Transportation Systems Magazine. 2023 Jan 1;15(1):136–59. doi: 10.1109/MITS.2022.3162901

[4] Lv Z, Li J, Dong C, Xu Z. DEEPSTF: A Deep Spatial–Temporal forecast model of taxi flow. The Computer Journal. 2021 Nov 16;66(3):565–80. doi: 10.1093/comjnl/bxab178

[5] Wang J, Yamamoto T, Li K. Spatial dependence and spillover effects in customized bus demand: Empirical evidence using spatial dynamic panel models. Transport Policy. 2021 May 1;105:166–80. doi: 10.1016/j.tranpol.2021.03.004

[6] Omonov F. The important role of intellectual transport systems in increasing the economic efficiency of public transport services. Academic Research in Educational Sciences. 2022;3(3). doi: 10.24412/2181-1385-2022-3-36-40

[7] Lv Z, Wang X, Chen Z, Li J, Li H, Xu Z. A new approach to COVID-19 data mining: A deep spatial–temporal prediction model based on tree structure for traffic revitalization index. Data and Knowledge Engineering. 2023 Jul 1;146:102193. doi: 10.1016/j.datak.2023.102193

[8] You L, Guan Z, Li N, Zhang J, Cui H, Claramunt C, et al. A Spatio-Temporal Schedule-Based neural network for urban taxi waiting time prediction. ISPRS International Journal of Geo-information. 2021 Oct 15;10(10):703. doi: 10.3390/ijgi10100703

[9] González SS, Bedoya-Maya F, Calatayud A. Understanding the effect of traffic congestion on accidents using big data. Sustainability. 2021 Jul 5;13(13):7500. doi: 10.3390/su13137500

[10] Xu Z, Lv Z, Li J, Shi A. A Novel Approach for Predicting Water Demand with Complex Patterns Based on Ensemble Learning. Water Resources Management. 2022 Jul 29;36(11):4293–312. doi: 10.1007/s11269-022-03255-5

[11] Lv Z, Li J, Dong C, Li H, Xu Z. Deep learning in the COVID-19 epidemic: A deep model for urban traffic revitalization index. Data and Knowledge Engineering. 2021 Sep 1;135:101912. doi: 10.1016/j.datak.2021.101912

[12] Fifty C, Amid E, Zhao Z, Yu T, Anil R, Finn C. Efficiently identifying task groupings for Multi-Task learning. arXiv (Cornell University). 2021 Sep 10; Available from: https://arxiv.org/pdf/2109.04617.pdf

[13] Jia H, Zhao J, Arshaghi A. COVID-19 Diagnosis from CT Images with Convolutional Neural Network Optimized by Marine Predator Optimization Algorithm. BioMed Research International. 2021 Oct 12;2021:1–9. doi: 10.1155/2021/5122962

[14] Asif M, Mitrovic N, Dauwels J, Jaillet P. Matrix and tensor based methods for missing data estimation in large traffic networks. IEEE Transactions on Intelligent Transportation Systems. 2016 Jul 1;17(7):1816–25. doi: 10.1109/tits.2015.2507259

[15] Gao P, Hu J, Zhou H, Zhang Y. Travel time prediction with immune genetic algorithm and support vector regression. 12th World Congress on Intelligent Control and Automation (WCICA). 2016 Jun 1. doi: 10.1109/wcica.2016.7578434

[16] Jenelius E, Koutsopoulos HN. Travel time estimation for urban road networks using low frequency probe vehicle data. Transportation Research Part B-methodological. 2013 Jul 1;53:64–81. doi: 10.1016/j.trb.2013.03.008

[17] Xiao Y, Qom SF, Hadi M, Al-Deek H. Use of Data from Point Detectors and Automatic Vehicle Identification to Compare Instantaneous and Experienced Travel Times. Transportation Research Record. 2014 Jan 1. doi: 10.3141/2470-10

[18] Zhang Z, Wang Y, Chen P, He Z, Yu G. Probe data-driven travel time forecasting for urban expressways by matching similar spatiotemporal traffic patterns. Transportation Research Part C-emerging Technologies. 2017 Dec 1;85:476–93. doi: 10.1016/j.trc.2017.10.010

[19] Osman O, Rakha HA, Mittal A. Application of Long Short Term Memory Networks for Long- and Short-term Bus Travel Time Prediction. Preprints. 2021 Apr 9. doi: 10.20944/preprints202104.0269.v1

[20] Lin WH, Wei H, Nian D. Integrated ANN-Bayes-based travel time prediction modeling for signalized corridors with probe data acquisition paradigm. Expert Systems With Applications. 2022 Dec 1;209:118319. doi: 10.1016/j.eswa.2022.118319

[21] Xu Z, Lv Z, Li J, Sun H, Sheng Z. A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. IEEE Intelligent Transportation Systems Magazine. 2023 Jan 1;15(1):136–59. doi: 10.1109/mits.2022.3162901

[22] Liu Y, Ren J, Ye J, Qu X. How machine learning informs ride-hailing services: A survey. Communications in Transportation Research. 2022 Dec 1;2:100075. doi: 10.1016/j.commtr.2022.100075

[23] Wang D, Zhang J, Cao W, Li J, Zheng Y. When Will You Arrive? Estimating Travel Time Based on Deep Neural Networks. Thirty-Second AAAI Conference on Artificial Intelligence. 2018 Apr 26;32(1). doi: 10.1609/aaai.v32i1.11877

[24] Qiu J, Du L, Zhang D, Su S, Tian Z. NEI-TTE: Intelligent Traffic Time Estimation based on Fine-Grained Time Derivation of road segments for Smart City. IEEE Transactions on Industrial Informatics. 2020 Apr 1;16(4):2659–66. doi: 10.1109/tii.2019.2943906

[25] Zhang J, Xiang Y, Wang Y, Zhou W, Xiang Y, Guan YL. Network traffic classification using correlation information. IEEE Transactions on Parallel and Distributed Systems. 2013 Jan 1;24(1):104–17. doi: 10.1109/tpds.2012.98

[26] Lan W, Xu Y, Zhao B. Travel Time Estimation without Road Networks: An Urban Morphological Layout Representation Approach. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. 2019 Aug 1. doi: 10.24963/ijcai.2019/245

[27] Zou ZT, Yang H, Zhu AX. Estimation of travel time based on ensemble method with Multi-Modality Perspective Urban Big data. IEEE Access. 2020 Jan 1;8:24819–28. doi: 10.1109/access.2020.2971008

[28] Shen L, Liu Y, Qu X. Model controlled prediction: a reciprocal alternative of model predictive control. IEEE/CAA Journal of Automatica Sinica. 2022 Jun 1;9(6):1107–10. doi: 10.1109/jas.2022.105611

[29] Tang K, Chen S, Liu Z. Citywide Spatial-Temporal Travel Time Estimation using big and sparse trajectories. IEEE Transactions on Intelligent Transportation Systems. 2018 Dec 1;19(12):4023–34. doi: 10.1109/tits.2018.2803085

[30] Nilashi M, Abumalloh RA, Alrizq M, Almulihi A, Alghamdi OA, Farooque MMJ, et al. A hybrid method to solve data sparsity in travel recommendation agents using fuzzy logic approach. Mathematical Problems in Engineering. 2022 Jun 27;2022:1–20. doi: 10.1155/2022/7372849

[31] Akhtar M, Moridpour S. A review of traffic congestion prediction using Artificial Intelligence. Journal of Advanced Transportation. 2021 Jan 29;2021:1–18. doi: 10.1155/2021/8878011

[32] Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning. Neurocomputing. 2021 Sep 1;452:48–62. doi: 10.1016/j.neucom.2021.03.091

[33] Ha KN, Cho S, MacLachlan DL. Response models based on bagging neural networks. Journal of Interactive Marketing. 2005 Feb 1;19(1):17–30. doi: 10.1002/dir.20028

[34] Li L, Dai S, Cao Z, Hong J, Jiang S, Yang K. Using improved gradient-boosted decision tree algorithm based on Kalman filter (GBDT-KF) in time series prediction. The Journal of Supercomputing. 2020 Jan 8;76(9):6887–900. doi: 10.1007/s11227-019-03130-y

[35] Abdollahi M, Khaleghi T, Yang K. An integrated feature learning approach using deep learning for travel time prediction. Expert Systems With Applications. 2020 Jan 1;139:112864. doi: 10.1016/j.eswa.2019.112864