



STTF: An Efficient Transformer Model for Traffic Congestion Prediction

Xing Wang^{1,2} · Ruihao Zeng³ · Fumin Zou⁴ · Lyuchao Liao⁴ · Faliang Huang⁵

Received: 21 October 2022 / Accepted: 18 December 2022
© The Author(s) 2023

Abstract

With the rapid development of economy, the sharp increase in the number of urban cars and the backwardness of urban road construction lead to serious traffic congestion of urban roads. Many scholars have tried their best to solve this problem by predicting traffic congestion. Some traditional models such as linear models and nonlinear models have been proved to have a good prediction effect. However, with the increasing complexity of urban traffic network, these models can no longer meet the higher demand of congestion prediction without considering more complex comprehensive factors, such as the spatio-temporal correlation information between roads. In this paper, we propose a traffic congestion index and devise a new traffic congestion prediction model spatio-temporal transformer (STTF) based on transformer, a deep learning model. The model comprehensively considers the traffic speed of road segments, road network structure, the spatio-temporal correlation between road sections and so on. We embed temporal and spatial information into the model through the embedding layer for learning, and use the spatio-temporal attention module to mine the hidden spatio-temporal information within the data to improve the accuracy of traffic congestion prediction. Experimental results based on real-world datasets demonstrate that the proposed model significantly outperforms state-of-the-art approaches.

Keywords Traffic congestion prediction · Free-stream velocity · Road network structure · Spatio-temporal information · Transformer

Abbreviations

STTF	Spatio-temporal transformer	MM	Markov models
TCI	Traffic Congestion Index	HMM	Hidden Markov models
TCS	Traffic Congestion Score	KNN	K-nearest neighbor
ARIMA	Autoregressive integrated moving average	DBSCAN	Density-based spatial clustering of applications with noise
EMD	Empirical mode decomposition	SVM	Support vector machine
		AMSVM	Adaptive multi-kernel SVM
		KELM	Kernel extreme learning machine
		S-ELM-Cluster	Symmetric-extreme learning machine cluster
		DTM	Decision tree models
		RF	Random forest
		BN	Bayesian network
		CNN	Convolutional neural network
		PCNN	Convolution-based deep neural network modeling periodic traffic data
		ASA-RGCNN	Analogous self-attention-residual gated CNN
		SG-CNN	Road segment group-CNN
		STFSA	Spatio-temporal feature selection algorithm
		LSTM	Long short-term memory
		RNN	Recurrent neural network

Xing Wang and Ruihao Zeng contributed equally to this work.

✉ Ruihao Zeng
rzen0142@uni.sydney.edu.au

¹ College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350108, China

² Digital Fujian Institute of Big Data Security Technology, Fuzhou 350108, China

³ School of Civil Engineering, The University of Sydney, Sydney 2008, Australia

⁴ Fujian Key Laboratory of Automotive Electronic and Electrical Drive Technology, Fujian University of Technology, Fuzhou 350117, Fujian, China

⁵ School of Computer and Information Engineering, Nanning Normal University, Nanning 530001, Guangxi, China

PrePCT	Predictor for position congestion tensor
GCN	Graph convolutional network
NGSIM	Next Generation SIMulation
AJB	Amended Jarque-Bera
NLP	Natural language processing
CV	Computer visual
DCRNN	Diffusion convolutional RNN
ST-GCN	ST-graph convolution network
MAE	Mean absolute error
RMSE	Root mean squared error
MAPE	Mean absolute percentage error

1 Introduction

In the past decade, with the rapid growth of global population and the acceleration of urbanization, cities have become more and more crowded, and urban road traffic is inevitably facing the problem of traffic congestion. Traffic congestion not only leads to inefficient transportation, but also increases the time and money spent by travelers. The environmental pollution problems are also aggravated by the increasing emissions of vehicles. Therefore, it is considered to be one of the important tasks for municipal management to solve the traffic congestion problem efficiently.

Current research on traffic congestion prediction can be mainly divided into three directions: linear models [9–15], nonlinear models [16–22] and neural network models [23–38]. Among them, linear models usually consider the traffic prediction values in a probability distribution model and make predictions by calculating the variation pattern of the predicted values on the timeline, for example, in literature [12, 15]. However, this type of models does not consider the spatio-temporal correlation between roads at all. The quantified road congestion is not a simple time-flow prediction problem. Considering that the general activity habits of most residents are regular, and roads may show the same changes at different times or different roads may show the same changes at the same time, all these potential relationships may help us to make better congestion prediction. Nonlinear models are mainly based on clustering and classification models, where researchers work to simplify complex flow changes into several different types of patterns and use them as a benchmark, such as in literature [16, 19]. But again, such type of models suffers from a lack of applicability. Considering the unsupervised nature of clustering, the optimal clustering criteria may also be completely different in areas with very different traffic conditions. Neural network models are widely used in congestion prediction because of their strong learning and in-depth mining ability for large-scale datasets, for example, in literature [24, 30]. But since traffic flow and road network structure are two completely different types

of information, it is difficult for the neural network to learn both features at the same time. Of course, some scholars have tried, for example, in literature [49, 50], to integrate the road network structure information into the graph network and learn it at the same time. However, the prediction accuracy still needs to be further improved. The model also needs to consider more critical impact factors, such as traffic flow, speed, running time, spatial and temporal correlation between road segments, etc.

Based on the above problems in traffic congestion prediction, we propose a new traffic congestion index with the introduced free-stream velocity of the road segment to reflect the road capacity and devise our prediction model Spatio-temporal Transformer (STTF).

Although there are many traffic data, such as traffic volume, vehicle speed and travel time that can reflect traffic congestion to some extent, the reason why we use free-stream velocity instead of traffic volume is that there are large gaps between main roads and non-main roads on traffic conditions in cities (especially large and densely populated cities). For example, traffic volume and speed are closely related to the geographical location of the road and the capacity of the road. A high traffic volume may only mean that the road segment is busy and does not necessarily indicate congestion. A low traffic volume may not necessarily indicate congestion if it is surrounded by residences or schools which has complex road conditions or has speed limits. Only traffic volume or speed does not accurately reflect the congestion of the road. Thus, we introduce the free-stream velocity to reflect the capacity of the road and then propose a new traffic congestion index.

besides, we deeply excavate the relationship among road network structure, correlation between road segments and road itself from spatio-temporal perspective. We take the construction of road network structure as the starting point and use the improved transformer to gradually retain the spatio-temporal information of roads.

The main contributions of this paper are summarized as follows.

- We propose a new traffic congestion index, which can accurately reflect the congestion degree of the road section according to the different traffic capacity and daily traffic conditions of each road.
- We devise an efficient STTF model for traffic congestion prediction based on the Transformer model, which can learn both the spatio-temporal information and road network structure information.
- We introduce an embedding learning module to learn the spatial and temporal information of the road network. On top of that, we encode and decode these two parts of information separately in the training phase to ensure that the model can obtain the spatio-temporal relationship of the data.

- In experiments with real-world datasets, our model has superior performance and accuracy compared with both classical and state-of-the-art models.

The remainder of paper is organized as follows. Section 2 mainly introduces the state-of-the-art research on traffic congestion prediction. Sections 3 and 4 mainly introduce the notations used in this paper and present our proposed model. In Sect. 5, we verify the superiority of STTF model by experiments. Finally, this paper is concluded in Sect. 6.

2 Related Work

Traffic congestion prediction can usually be viewed as a complex time series prediction problem. Considering the rich variety of data in the traffic domain and referring to Akhtar et al. [1] for an overview of research in this area, we can classify the research directions into direct and indirect types based on the type of data. Among them, the direct type of methods uses data that may affect traffic conditions, such as weather conditions [2] and emergencies [3], which often give direct information about the traffic status and facilitate drivers' judgments. Some data that reflect the public state can also directly reflect the congestion, such as the diversion structure of roads [4], public opinion reports [5], and electricity consumption [6].

The indirect type of methods is the one that has been studied by more scholars. These methods usually use some vehicle travel data, such as traffic flow, vehicle speed, etc. Although these data do not directly reflect the traffic congestion information, researchers use these basic data to quantify the congestion as a parameter [7, 8], which is usually called Traffic Congestion Index (TCI) or Traffic Congestion Score (TCS) and forecast traffic congestion with TCI or TCS predictions. We can simply classify this type of research into three categories, which are linear models, nonlinear models, and neural network models.

2.1 Linear Models

Linear model-based approaches usually consider traffic data to satisfy a particular distribution. Such approaches include traditional mathematical statistical models and state-space models. Traditional statistical models were first used for traffic state prediction by Nicholson et al. [9] who used spectral analysis to find the interconnections of data in the time dimension. Later, Yang et al. [10] used road occupancy, He et al. [11] used speed performance index to mine road congestion probability are similar reasoning. Besides, in recent years, Autoregressive Integrated Moving Average (ARIMA) model is also widely used in the research. For example, Alghamdi et al. [12] used ARIMA model to

study the factors affecting traffic congestion and proposed a short-term prediction model for non-Gaussian distributed data. Wang et al. [13] combined ARIMA with Empirical Mode Decomposition (EMD), based on which the hybrid framework has better short-term prediction than similar methods. In addition, methods based on Markov Models (MM) and Hidden Markov Models (HMM) are also widely used. For example, Zaki et al. [14] used HMM to find a suitable Neuro-Fuzzy prediction network for congestion at a specific period, while Ali-Eldin et al. [15] used HMM to construct a two-dimensional space based on average speed and contrast and used it to capture the changing patterns of traffic conditions.

This type of linear time-series-based prediction models usually utilizes only the temporal characteristics of the data and does not consider other additional information. So, it is only suitable for road data with strong stability in the time dimension.

2.2 Nonlinear Models

With the increasing randomness and volatility of modern urban traffic, it is difficult for simple linear models to meet the requirements for congestion prediction. Therefore, researchers have started to use non-linear models to tap into traffic variations.

One of the main categories is the mining of traffic patterns from the perspective of historical data using clustering models led by K-Nearest Neighbor (KNN) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). For example, Wen et al. [16] used DBSCAN to find the spatio-temporal association rules of roads and performed classification simulation for different patterns of road links to improve the prediction accuracy. Support Vector Machine (SVM) is also widely used for congestion prediction due to its non-linear regression capability. For example, Feng et al. [17] proposed an Adaptive Multi-kernel SVM (AMSVM) using Gaussian kernel and polynomial kernel to explore the stochasticity and spatio-temporal relationship of traffic flow. Xing et al. [18] proposed a Kernel Extreme Learning Machine (KELM) based on kernel function as the replacement of hidden layer, while Ban et al. [19] proposed an efficient learning method based on Symmetric-Extreme Learning Machine Cluster (S-ELM-Cluster), which is able to transform large-scale data learning to different problems on small-scale datasets. In addition, Decision Tree Models (DTM) [20], Random Forest (RF) [21] and Bayesian Network (BN) [22] also have similar ideas to nonlinear methods.

2.3 Neural Network Models

With the development of deep learning, neural network models [23] have achieved excellent results in more and

more prediction fields, no exception in traffic congestion prediction.

Among them, Convolutional Neural Network (CNN) has been chosen by many researchers for its powerful feature extraction capability and adaptability to high-dimensional data. For example, Zhu et al. [24] used CNN to detect complex traffic conditions in Bath City. Chen et al. [25] proposed a PCNN (Convolution-based deep Neural Network modeling Periodic traffic data) model capable of transforming one-dimensional data into image-like data for input. Zhang et al. [26] proposed an Analogous Self-Attention-Residual Gated CNN (ASA-RGCNN) model combining gated structure and ASA structure to discover the impact of data spatio-temporal characteristics on different levels of traffic flow. Meanwhile, CNNs are usually combined with graph networks. SG-CNN (Road Segment Group-CNN) proposed by Tu et al. [27] can mine the common information between road segments, while the work of Zhang et al. [28] that used Spatio-Temporal Feature Selection Algorithm (STFSA) to extract spatio-temporal information and then handed over to CNN for learning, which had also been proved to have better prediction accuracy. In addition, the Long Short-Term Memory (LSTM) [29] network adapted from Recurrent Neural Network (RNN) is often used as a benchmark method for prediction because of its temporal learning capability. For example, Bai et al. [30] used LSTM to learn temporal features while using Predictor for Position Congestion Tensor (PrePCT) and CNN for spatial features. Similar ideas are used in literature [31–34].

Besides, some emerging models such as Graph Convolutional Network (GCN) [35], Next Generation Simulation (NGSIM) [36], MetaNet [37], and Attention mechanism [38] are able to perform high accuracy congestion prediction. However, neural network models still have some urgent problems, such as the loss of the ability to mine hidden information because of the depth of the network, the high consumption of computational resources, and the inability to consider the road network structure information completely.

3 Notations

In this section, we focus on some of the basic parameters that will be used in this paper.

3.1 Traffic Congestion Index

According to the description in Sect. 1, we introduce the vehicle free-stream velocity $v_{\text{free-stream}}$ and propose a new traffic congestion index TCI . we denote the current time period as t , and use \hat{v}_t to denote the average velocity of all

sampled vehicles passing through one road segment or sensor in time period t . Then TCI is calculated as follows,

$$TCI_t = \begin{cases} 1 - \frac{\hat{v}_t}{v_{\text{free-stream}}}, & \hat{v}_t \leq v_{\text{free-stream}}, \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where $v_{\text{free-stream}}$ is defined as the speed of a vehicle passing the road segment under ideal conditions (only one vehicle on the road and no external factors are considered), i.e., the maximum withstand speed of the road. By observing the vehicle speed distribution in the public dataset PeMS-Bay, Beijing and the private dataset FUZHOU we can simply assume that the vehicle speed variable v conforms to the normal distribution $\mathcal{N} = (\mu, \sigma^2)$. Then the probability density function $f(v)$ of the speed is as follows.

$$f(v) = \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp\left(-\frac{(v - \mu)^2}{2\sigma^2}\right). \tag{2}$$

Then, if this assumption holds, the variables should satisfy the more efficient amended Jarque-Bera (AJB) test [27], i.e.,

$$AJB = \frac{(VS)^2}{\text{Var}(VS)} + \frac{(VK - E(VK))^2}{\text{Var}(VK)}, \tag{3}$$

where VS is velocity skewness, which is used to measure the direction and degree of skewness of the velocity sample data distribution. VS is represented by the third-order standard matrix of the velocity variable v . The VK is velocity kurtosis, which is used to indicate the sharpness of the peak of the velocity sample data distribution. VK is represented by the fourth-order standard matrix of the velocity variable v . The two variables are calculated as follows.

$$VS = E\left[\left(\frac{v - \mu}{\sigma}\right)^3\right], \tag{4}$$

$$VK = E\left[\left(\frac{v - \mu}{\sigma}\right)^4\right]. \tag{5}$$

In the case of satisfying the AJB test we can express the free flow speed $v_{\text{free-stream}}$ of vehicle in terms of the total overall expectation, i.e.,

$$v_{\text{free-stream}} = \int v \cdot f(v)dv. \tag{6}$$

3.2 Road Network Structure Graph

We first denote the selected road network as a weighted directed graph \mathcal{G} where $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$. Among it, \mathcal{V} is the set of nodes in the network and the number of vertices $N = |\mathcal{V}|$. \mathcal{E}

is the set of link states in the network, and \mathcal{W} is the set of weights between nodes, which can be regarded as a weight matrix and $\mathcal{W} \in \mathbb{R}^{N \times N}$. Therefore, \mathcal{W}_{v_i, v_j} denotes the link weights between nodes v_i and v_j . The exact calculation method will be given later. In the traffic road network, each node represents a specific road segment (or sensor), while the link between nodes indicates the connected relationship between road segments (or sensors). The link weights represent the degree of association between connected road segments (or sensors).

Then the time variables are defined. We define historical time steps as h and future time steps as f . Then the TCI_t in the current time period t can be represented by the matrix X_t and $X_t \in \mathbb{R}^N$. Therefore, the TCI of the road network \mathcal{V} in the past time period h is denoted as $\mathcal{X} = (X_{t1}, X_{t2}, \dots, X_{th})$ and $\mathcal{X} \in \mathbb{R}^{N \times h}$, while the TCI of the future f time steps that need to be predicted can be written as $\mathcal{Y} = (X_{th+1}, X_{th+2}, \dots, X_{th+f})$ and $\mathcal{Y} \in \mathbb{R}^{N \times f}$. The ground truth of the TCI in the future f time steps can be written as $\mathcal{U} = (U_{th+1}, U_{th+2}, \dots, U_{th+f})$.

In that case, we give a definition of the link weight coefficient \mathcal{W} so that it reflects the actual distance between two interconnected road segments and the correlation between the two road segments. Then we have the following equation,

$$\mathcal{W}_{v_i, v_j} = \begin{cases} \exp\left(-\frac{r(v_i, v_j)}{d_{v_i, v_j} / \xi}\right), & \exp\left(-\frac{r(v_i, v_j)}{d_{v_i, v_j} / \xi}\right) \geq \varepsilon, \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where d_{v_i, v_j} denotes the actual distance between the center points of the two road segments. $r(v_i, v_j)$ denotes the Pearson Correlation Coefficient of the traffic flow at nodes v_i and v_j . Here, ξ is introduced as the adjustment factor to make d_{v_i, v_j} and $r(v_i, v_j)$ comparable, which is taken as $\xi = 1000$. ε is the threshold used to control the degree of \mathcal{W} diffusion, and here $\varepsilon = 0.05$.

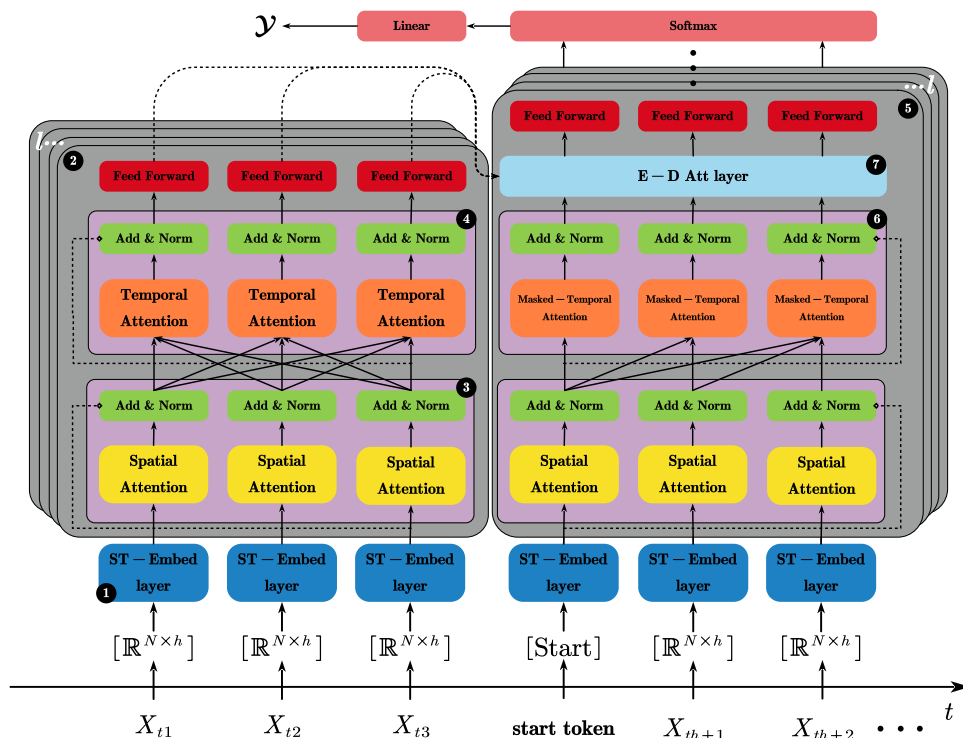
4 STTF Model

In this section, we introduce the structure of the proposed STTF model and the functions of each part.

4.1 The STTF Model Structure

The Transformer model was first proposed by Vaswani et al. [39]. The Attention mechanism, encoder and decoder in the model together form the black box, which is the core structure of this model. The complex nature of its parallelized computation dictates that it is better than RNN in terms of accuracy and performance. Transformer has previously been widely used in the Natural Language Processing (NLP) [40] and Computer Visual (CV) [41] fields. Lim et al. [42] also used it to mine the temporal dimensional features of time series data, but studies using Transformer to mine spatio-temporal patterns are less common.

Fig. 1 Structure of STTF model



Based on the classical Transformer, we propose a new Spatio-Temporal Transformer (STTF) model. The complete structure is shown in Fig. 1. The Transformer framework mainly consists of encoder, decoder and embedding module, which contains the new given ST-Embedding layer (Spatial Embed block & Temporal Embed block), the new given ST-Attention layers (Spatial Att layer & Temporal Att layer) and other classical structures. The input of the Transformer is the *TCI* data \mathcal{X} at h time steps in the past. The output is the predicted *TCI* data \mathcal{Y} at f time steps in the future. Each module is set to output a D -dimensional vector to facilitate the connection of the modules in each layer.

4.2 ST-Embedding Layer

ST-Embedding layer is the number 1 module in Fig. 1.

Spatial Embed block Considering that the road network structure graph \mathcal{G} is a directed acyclic graph with weights, to transform it into variables that the Transformer can learn and retain the structural information, we need to transform the network nodes into vector form represented in the vector space. Here we use the LINE algorithm proposed by Tang et al. [43]. The input structure graph \mathcal{G} is vectorically represented and a feedforward neural network with GRLU activation function is added after the output to transform it into a D -dimensional vector. Then the final output is noted as se_{v_i} while $se_{v_i} \in \mathbb{R}^D, v_i \in \mathcal{V}$.

Temporal Embed block Spatial Embed block provides structural information of road data, then Temporal Embed block is also needed to provide temporal feature information for Transformer. Referring to the nonlinear method to learn the distribution pattern of data in time dimension by historical data, the historical data is also used here for embedding encoding. Considering the uniqueness of each time dimension, one-hot encoding [44] is used here to encode the time in the past h steps. We encode the number of days in a week into the vector space of \mathbb{R}^7 and the time period t in a day into the vector space of \mathbb{R}^t . Finally, the two encodings are transformed into the vector space of \mathbb{R}^{7+t} by concatenation operation, and a feedforward neural network with GRLU activation function is also added to the output to transform it into a D -dimensional vector. In this case, we can encode the temporal features of the past h time steps and write the vector of the neural network output as te_{t_j} while $te_{t_j} \in \mathbb{R}^D, t_j \in \{t_1, t_2, \dots, t_h\}$.

After getting the feature information of temporal embedding and spatial embedding respectively, we need to integrate the two parameters of the same dimensions. Here we introduce the new embedding coefficients ste_{v_i, t_j} , and we can get the following embedding coefficients in t_j steps of node v_i .

$$ste_{v_i, t_j} = \exp \left(- \left(se_{v_i} + te_{t_j} \right) \right). \tag{8}$$

We denote this operation as \odot . Then the ST-Embedding layer structure diagram is shown below in Fig. 2.

4.3 Encoder Architecture

Encoder is the number 2 module in Fig. 1. A total of L encoders are included in STTF model. Each encoder consists of three consecutive layers: Spatial Att layer (number 3 module in Fig. 1), Temporal Att layer (number 4 module in Fig. 1), and Feed Forward layer (where Spatial Att layer and Temporal Att layer together form the ST-Attention layer). The first two attention structures have a skip-connection structure used to skip inter-layer connections (indicated by dashed lines). To improve the generalization ability, each attention operation is employed the normalization and dropout. The Feed Forward layer is mainly designed to integrate high-dimensional attention information and consists of two fully connected neural networks with ReLU activation functions. After feeding the feature vector sequence \mathcal{X} to the first encoder, the ST-Embedding layer finally outputs the hidden representation vector of the encoder to the decoder's attention layer after the $L - 1$ encoder's attention operation.

Referring to the design of the attention layer in the classical Transformer structure [39], we propose a two-layer ST-Attention layer structure consisting of Spatial Att layer and Temporal Att layer. Each encoder and decoder has one ST-Attention layer, then we can note that in l -th ST-Attention layer in encoder or decoder, the output of Spatial Att layer is $sa_{v_i, t_j}^{(l)}$, and the output of Temporal Att layer is $ta_{v_i, t_j}^{(l)}$. Then there are l -th ST-Attention layer whose input is $z_{v_i, t_j}^{(l-1)}$ and output is $z_{v_i, t_j}^{(l)}$.

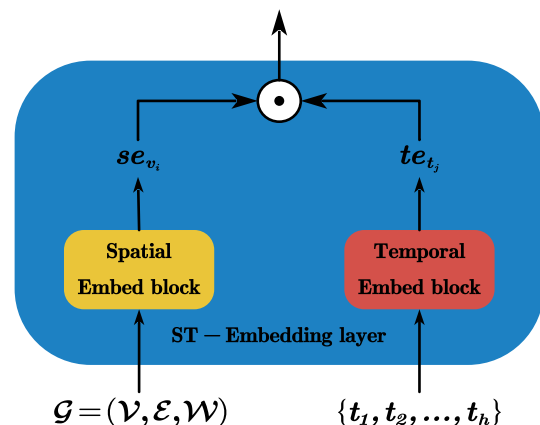


Fig. 2 Structure of ST-Embedding layer

Spatial Att layer To fully consider the influence of each road link on the specified road segment in the road network structure, we calculate the effect of each node in the $(l - 1)$ -th Spatial Att layer on the node v_i in the l -th layer, i.e., assign different weights to each node at different time periods, which is shown in Fig. 3. Then the output hidden representation vector of this layer is calculated below,

$$sa_{v_i,t_j}^{(l)} = \sum_{v \in \mathcal{V}} \left(\alpha_{v_i,v} \cdot z_{v,t_j}^{(l-1)} \right), \tag{9}$$

where $\alpha_{v_i,v}$ denotes the normalized attention coefficient. Noting its pre-normalization state as $sr_{v_i,v}$, which is directly used to represent the correlation coefficient between each node v of upper layer and the given node v_i of current layer. According to the classical Transformer structure [39], we choose to use scaled dot-product approach to represent the correlation between the two nodes. Then we can obtain the following equation,

$$sr_{v_i,v} = \frac{\left[\text{Concat} \left(z_{v_i,t_j}^{(l-1)}, \text{ste}_{v_i,t_j} \right), \text{Concat} \left(z_{v,t_j}^{(l-1)}, \text{ste}_{v,t_j} \right) \right]}{\sqrt{d}}, \tag{10}$$

where $[a, b]$ denotes the calculation of the inner product of a and b . d denotes the dimension of the vector after the concatenation operation is performed. Thus, we normalize $sr_{v_i,v}$ using the softmax function to obtain $\alpha_{v_i,v}$.

$$\alpha_{v_i,v} = \frac{\exp \left(sr_{v_i,v} \right)}{\sum_{v \in \mathcal{V}} \exp \left(sr_{v_i,v} \right)}. \tag{11}$$

Finally, to improve the efficiency and expand the capacity of the network through parallel computation, we introduce the multi-head attention mechanism [39]. We set the number of attention heads to Q , i.e., use different, learnable linear

projections to project each parameter linearly Q times to the corresponding dimension. The attention function of each projection is computed in parallel, and the concatenation operation is performed after each computation. In that case, we denote the projection operation as p . Then, $p(x)$ is the linear projection function, which is calculated below,

$$p(x) = \text{sigmoid}(mx + n), \tag{12}$$

where both m and n denote learnable variable parameters. $p_{m,n}^{(h)}$ denotes the projection function with different parameters. Then these can be obtained that at the q -th projection,

$$\alpha_{v_i,v}^{(q)} = \frac{\exp \left(sr_{v_i,v}^{(q)} \right)}{\sum_{v \in \mathcal{V}} \exp \left(sr_{v_i,v}^{(q)} \right)} \tag{13}$$

$$sr_{v_i,v}^{(q)} = \frac{\left[p_{m,n}^{(q)} \left(\text{Concat} \left(z_{v_i,t_j}^{(l-1)}, \text{ste}_{v_i,t_j} \right) \right), p_{m,n}^{(q)} \left(\text{Concat} \left(z_{v,t_j}^{(l-1)}, \text{ste}_{v,t_j} \right) \right) \right]}{\sqrt{\frac{d}{q}}}, \tag{14}$$

$$sa_{v_i,t_j}^{(l,q)} = \text{Concat}_{Q=1}^q \left(\sum_{v \in \mathcal{V}} \left(\alpha_{v_i,v}^{(q)} \cdot p_{m,n}^{(q)} \cdot z_{v,t_j}^{(l-1)} \right) \right). \tag{15}$$

Temporal Att layer To fully explore the hidden temporal patterns in the historical data of the same road segment, Temporal Att layer is introduced in encoder, whose input is the output $sa_{v_i,t_j}^{(l)}$ of Spatial Att layer of the same ST-Attention layer. We calculate the influence of past and future moment of node v_i in each Temporal Att layer on the present moment, which is shown in Fig. 4. Using the same computational model and time vector as in the Spatial Att layer, the hidden representation vector of the layer output is noted as ta_{v_i,t_j} , the attention coefficient in the layer is denoted by $\beta_{t_i,t}$, and its

Fig. 3 Principle of spatial attention

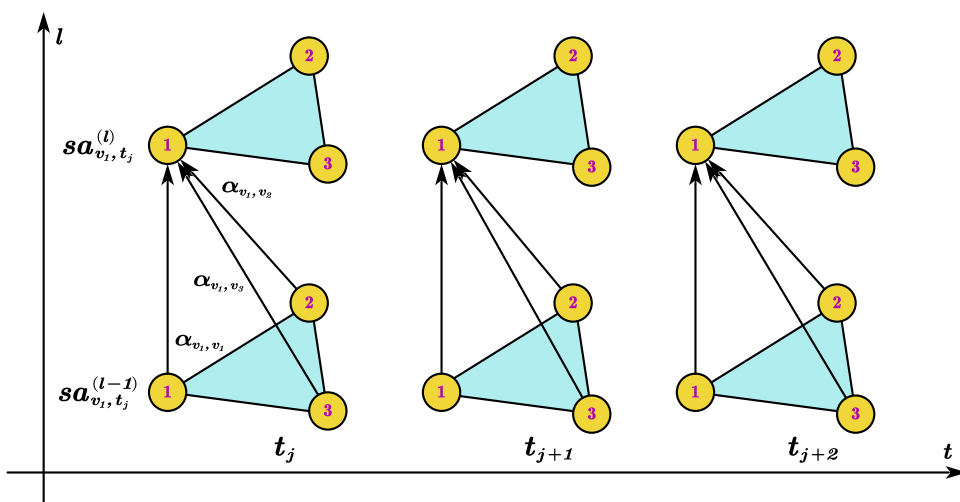
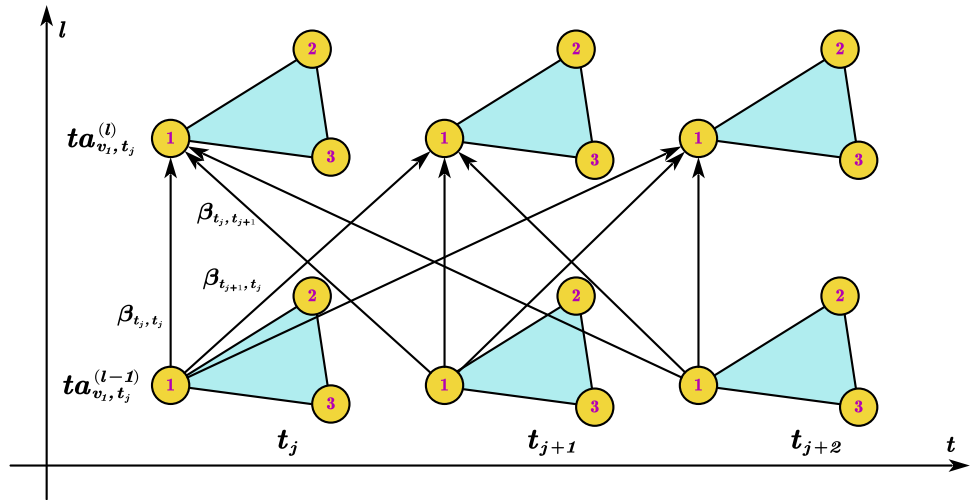


Fig. 4 Principle of temporal attention



state before normalization is denoted as $tr_{t_j, t}$, which indicates the impact of t time step on the current step t_j of same road segment. The hf_{t_j} is the set of all time steps before and after the step t_j (including the current step t_j). Finally, the multi-head attention mechanism, $p_{m,n}^{(h)}$, is introduced to denote the projection function with different parameters. Then we have the following equations.

$$tr_{t_j, t} = \frac{\left[\text{Concat} \left(sa_{v_i, t_j}^{(l)}, ste_{v_i, t_j} \right), \text{Concat} \left(z_{v_i, t_j}^{(l)}, ste_{v_i, t_j} \right) \right]}{\sqrt{d}}, \quad (16)$$

$$\beta_{t_j, t} = \frac{\exp \left(tr_{t_j, t} \right)}{\sum_{t \in hf_{t_j}} \exp \left(tr_{t_j, t} \right)}, \quad (17)$$

$$ta_{v_i, t_j}^{(l)} = \sum_{t \in hf_{t_j}} \left(\beta_{t_j, t} \cdot z_{v_i, t_j}^{(l-1)} \right). \quad (18)$$

Then these can be obtained that at the q -th linear projection,

$$tr_{t_j, t}^{(q)} = \frac{\left[p_{m,n}^{(q)} \left(\text{Concat} \left(sa_{v_i, t_j}^{(l)}, ste_{v_i, t_j} \right) \right), p_{m,n}^{(q)} \left(\text{Concat} \left(sa_{v_i, t_j}^{(l)}, ste_{v_i, t_j} \right) \right) \right]}{\sqrt{\frac{d}{q}}}, \quad (19)$$

$$\beta_{t_j, t}^{(q)} = \frac{\exp \left(tr_{t_j, t}^{(q)} \right)}{\sum_{t \in hf_{t_j}} \exp \left(tr_{t_j, t}^{(q)} \right)}, \quad (20)$$

$$ta_{v_i, t_j}^{(l, q)} = \text{Concat}_{Q=1}^q \left(\sum_{t \in hf_{t_j}} \left(\beta_{t_j, t}^{(q)} \cdot p_{m,n}^{(q)} \cdot sa_{v_i, t}^{(l)} \right) \right). \quad (21)$$

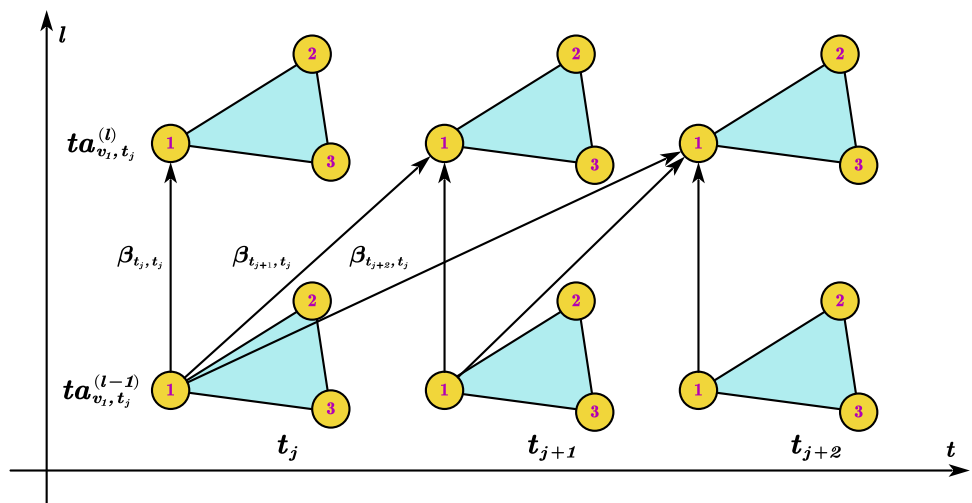
Masked-Temporal Att layer The Masked-Temporal Att layer (number 6 module in Fig. 1), exists only in the decoder. The only difference between it and the Temporal Att layer is that it masks the influence of future time steps on the present time step, thus limiting the attention of the decoder to the historical time steps, which is shown in Fig. 5. Therefore, by defining \mathcal{T}_{t_j} as the set of all time steps before the step t_j (including the current step t_j), we have the equation below.

$$ta_{v_i, t_j}^{(l, q)} = \text{Concat}_{Q=1}^q \left(\sum_{t \in \mathcal{T}_{t_j}} \left(\beta_{t_j, t}^{(q)} \cdot p_{m,n}^{(q)} \cdot sa_{v_i, t}^{(l)} \right) \right). \quad (22)$$

4.4 Decoder Architecture

Encoder is the number 5 module in Fig. 1. A total of L decoders are included in STTF model. The overall structure of each decoder is similar to that of the encoder, including an identical Spatial Att layer, an amended Masked-Temporal Att layer, a classical E-D Att layer (Encoder-Decoder Attention layer, number 7 module in Fig. 1) [39], and an identical Feed Forward layer. Among them, the E-D Att layer extracts feature information using encoder and Masked-Temporal Att layer's encoding vectors. Each node's embedding vector ste_{v_i, t_j} at future time steps and ste_{v_i, t_h} at historical time steps

Fig. 5 Principle of masked-temporal attention



correspond to the key and value in the classical structure, respectively. After decoder outputting the feature space vector, the prediction sequence \mathcal{Y} is finally outputted by linear layer and normalization operation.

5 Experiments

To test the practical effectiveness of our model, we conduct experiments on two real-world large-scale datasets, respectively.

5.1 Datasets

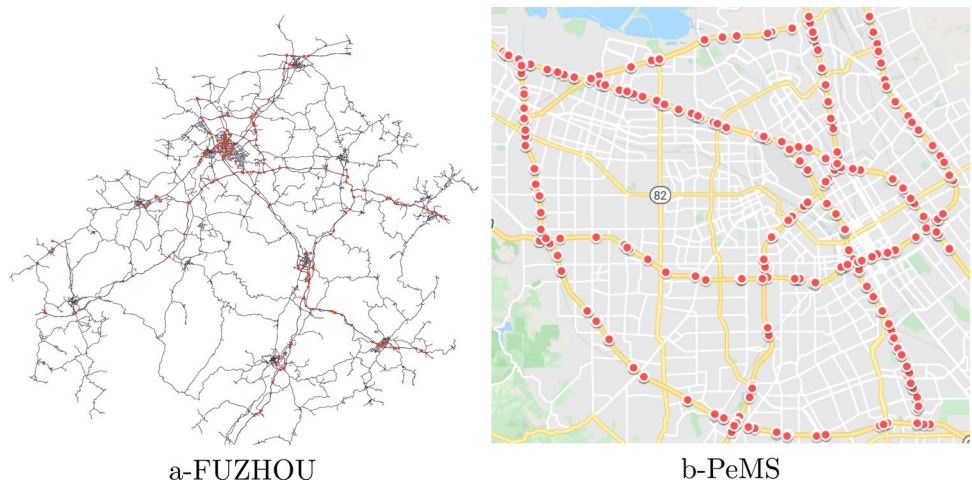
Considering that FUZHOU is vehicle GPS data and PeMS-Bay is sensor data, we first use the IVMM algorithm [45] to do map matching for the vehicles data in FUZHOU. After that, we count the speed data in both datasets in every 5,

10, and 15 minutes and fill the missing data with 0 values as well as normalized the data in the way of Li et al. [46].

FUZHOU This private traffic dataset is collected by Department of Transport of Fujian Province. The dataset contains speed data for 2 months ranging from May 1 st to June 31 st in 2018 , gathered from part of urban roads in Fuzhou City, Fujian Province. The distribution of road sections is shown in the following Fig. 6a.

PeMS-Bay This public traffic dataset is collected by California Transportation Agencies (Cal-Trans) Performance Measurement System (PeMS). The dataset contains speed data for 6 months ranging from January 1 st to May 31 st in 2017 from 325 sensor, gathered from highway in Bay Area, Los Angeles. The distribution of the sensors is shown in Fig. 6b below. Among them, considering the complexity of urban road links, we consider that the information complexity of FUZHOU dataset is higher than that of PeMS-Bay dataset.

Fig. 6 Dataset description (FUZHOU and PeMS-Bay)



5.2 Experimental Configuration

According to the method of Li et al. [46], we set a standard time step of 5 minutes. Thus, the historical time periods $h = 12$ time steps and the future time periods $f = 12$ time steps, i.e., both are one hour. For the use of optimizer, we choose Adam-warmup optimizer [47] and set the initial learning rate as 0.001, warmup step size and batch size as 4000 and 20, respectively.

In STTF model, there are three hyperparameters, namely, the number of layers L of Encoder and Decoder, the number of attention heads Q in the multi-head attention mechanism, and the vector dimension D of the output of each module. After several experiments and referring to the setting of the classic transformer structure, we selected the hyperparameter with the better performance, i.e., $L = 4$, $Q = 8$, and $D = 64$. In addition, we set the dropout rate to 0.3 and initialize the parameters of the network using Xavier weight initialization [48].

5.3 Baselines and Measures

We select five benchmark models for comparative experiments, including some basic models in the prediction problem and some state-of-the-art deep learning models. These five baselines are ARIMA [12], PrePCT [30], DCRNN (Diffusion Convolutional RNN) [46], ST-GCN (ST-Graph Convolution Network) [49], and Graph WaveNet [50]. Among them, ARIMA is the representative work in the linear model, PrePCT and DCRNN are the state-of-the-art convolutional neural network models, and the remaining two models are the state-of-the-art graph neural network models. Considering the different training mechanisms and the lack of labels, a comparison with the non-linear model is not made here. The codes of all the above models are publicly available by the authors, so we can all experiment with our own datasets. In our experiments, we measure the accuracy of the models by three widely used metrics, namely, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean

Absolute Percentage Error (MAPE). For a more visual comparison of values, all MAE and RMSE values are artificially expanded by a factor of 50.

5.4 Experimental Results and Discussion

The main purpose of our experiments is to explore the prediction accuracy, the generalization ability for different road conditions, the robustness under different time intervals and time steps, and the computational efficiency of the model. Therefore, we design several experiments to test STTF model by varying the time variables and road conditions. Moreover, the prediction time step indicates the time period of the model prediction results, the standard time step denotes the time period used in model learning, and the time interval indicates the time period of integrating data during data processing.

We first test the prediction accuracy of the model under different prediction time step. In Fuzhou dataset, complex road network structure data can better verify the prediction ability of each model. Then all six models are made to predict the change value of *TCI* every 30 min during the main weekday period (June 4, 2018, Monday, 6:00–20:00). The visualization results are shown in the Fig. 7 below, where ground truth is bolded. We can find that the STTF model has significantly stronger accuracy compared to the ARIMA and PrePCT models, especially for peak values and moments with large change rates that STTF is better fitted to the ground truth. To better compare quantitatively with the remaining deep neural network models, we calculate the MAE, RMSE, and MAPE values of the six models for the given time periods in the FUZHOU dataset under different prediction ranges. The results are shown in Table 1. From the results we can see that ARIMA performs the worst under the same prediction range because of its singularity of temporal characteristics. The prediction ability of PrePCT differs more from its authors' experimental results, probably because it is more suitable for road network prediction with a smaller number of nodes. The better performance of

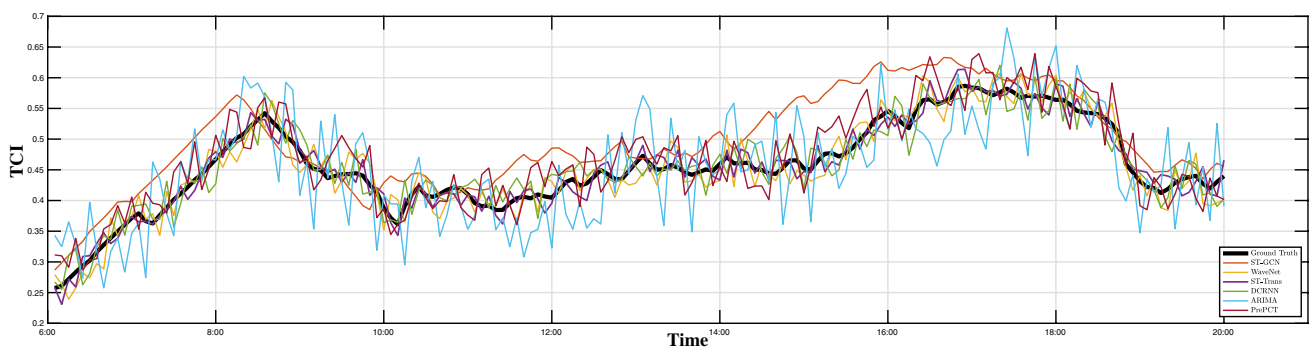


Fig. 7 *TCI* prediction results

Table 1 Performance of each model in FUZHOU database under different prediction time step (predictions are made every 30 minutes and the optimal values under the same time step are bolded)

	Prediction time steps	Metrics	ARIMA	PrePCT	DCRNN	ST-GCN	Graph WaveNet	STTF
FUZHOU	15 min	MAE ($\times 50$)	3.577	2.877	2.312	2.276	2.253	2.225
		RMSE ($\times 50$)	4.027	3.359	2.460	2.452	2.387	2.349
		MAPE	7.5%	6.1%	5.7%	5.9%	5.57%	5.52%
	30 min	MAE ($\times 50$)	3.974	3.111	2.407	2.397	2.416	2.358
		RMSE ($\times 50$)	4.895	3.442	2.517	2.518	2.463	2.382
		MAPE	7.5%	6.6%	5.8%	6.2%	5.63%	5.52%
	60 min	MAE ($\times 50$)	4.339	3.247	2.528	2.522	2.543	2.497
		RMSE ($\times 50$)	5.401	3.485	2.559	2.568	2.535	2.505
		MAPE	12.2%	10.2%	7.5%	8.4%	7.28%	7.24%

Table 2 Performance of each model in PeMS-BAY database under different prediction time step (predictions are made every 30 minutes and the optimal values under the same time step are bolded)

	Prediction time steps	Metrics	ARIMA	PrePCT	DCRNN	ST-GCN	Graph WaveNet	STTF
PeMS-Bay	15 min	MAE ($\times 50$)	1.573	0.874	0.305	0.288	0.271	0.263
		RMSE ($\times 50$)	2.053	1.337	0.435	0.442	0.389	0.362
		MAPE	3.4%	3.2%	2.8%	2.9%	2.79%	2.71%
	30 min	MAE ($\times 50$)	1.969	1.127	0.392	0.411	0.388	0.357
		RMSE ($\times 50$)	2.883	1.432	0.512	0.504	0.447	0.404
		MAPE	5.6%	4.7%	3.9%	4.1%	3.59%	3.51%
	60 min	MAE ($\times 50$)	2.334	1.276	0.539	0.548	0.514	0.529
		RMSE ($\times 50$)	3.418	1.508	0.580	0.574	0.536	0.577
		MAPE	8.8%	7.1%	4.7%	5.5%	4.32%	4.39%

graph-based deep learning models such as DCRNN illustrates that current deep learning methods are better than most traditional linear methods, and that neural networks based on graph structures are more likely to perform better than traditional time series networks. The STTF model outperforms all benchmark models, which proves that our ST-Attention layer can better mine hidden information and is more efficient compared to short-term serial prediction methods.

Second, we need to consider the performance of the models in road network structures with different levels of complexity, where the change of road complexity is reflected in the difference of data collection locations. Therefore, we test the prediction performance of *TCI* for each model in the selected time periods (March 6, 2017, Monday, 6:00–20:00) of the PeMS-Bay dataset in different prediction ranges. The results are presented in Table 2. Combining Tables 1 and 2 we can see that the STTF model outperforms most of the benchmark models, and its predictions are more stable for complex road networks. It only loses to Graph WaveNet in predicting *TCI* values for 60 min. Such a situation may be

explained by the following. In the complex road networks, each road segment has more neighboring road segments. One road segment may affect more road segments. More valid information is credited when we consider the impact of all road segments on the specific one road segment in the structure. Whereas in a relatively simple road structure, a road segment may only affect some neighboring road segments. When we record its impact on all other road segments, more invalid information enters the network, which eventually leads to different performance of the STTF model in the face of road networks of different complexity.

In addition, the traffic conditions in the same city may have exceptions during both peak/off-peak hours and weekdays/weekends, we further validate the STTF model's ability to cope with these exceptions.

Firstly, we consider that traffic volumes tend to have different patterns of variation at different times of the day, i.e., what we generally consider as morning peak, evening peak, and off-peak periods. This directly results in different peaks and different rates of change of *TCI* values for each time period. Therefore, we test the generalization

Table 3 Performance of each model with different time intervals and same prediction time steps at different time periods of the day (predictions are made every 30 minutes and the optimal values within the same time period are bolded)

Model - Time periods	15 min			30 min			60 min		
	MAE (×50)	RMSE (×50)	MAPE	MAE (×50)	RMSE (×50)	MAPE	MAE (×50)	RMSE (×50)	MAPE
DCRNN (4–12)	2.343	2.574	5.82%	2.532	2.595	5.78%	2.691	2.419	7.53%
ST-GCN (4–12)	2.471	2.422	5.93%	2.536	2.578	6.15%	2.617	2.483	8.56%
Graph WaveNet (4–12)	2.294	2.573	5.56%	2.544	2.355	5.69%	2.692	2.352	7.35%
STTF (4–12)	2.134	2.154	5.50%	2.498	2.561	5.64%	2.547	2.533	7.11%
DCRNN (12–20)	2.187	2.394	5.60%	2.393	2.413	5.62%	2.516	2.561	7.46%
ST-GCN (12–20)	2.166	2.377	5.88%	2.269	2.506	6.03%	2.422	2.496	8.30%
Graph WaveNet (12–20)	2.170	2.318	5.52%	2.253	2.289	5.49%	2.506	2.362	7.16%
STTF (12–20)	2.223	2.166	5.51%	2.217	2.222	5.48%	2.322	2.356	7.07%
DCRNN (20–4)	2.249	2.507	5.67%	2.355	2.536	5.85%	2.509	2.568	7.56%
ST-GCN (20–4)	2.287	2.402	5.97%	2.377	2.572	6.25%	2.555	2.578	8.47%
Graph WaveNet (20–4)	2.265	2.322	5.52%	2.366	2.507	5.63%	2.506	2.491	7.36%
STTF (20–4)	2.188	2.377	5.51%	2.324	2.346	5.49%	2.445	2.536	7.25%

Table 4 Performance of each model in weekdays and weekends with different time intervals and same prediction time steps (predictions are made every 30 minutes and the optimal values in the same day are bolded)

Model - Days	15 min			30 min			60 min		
	MAE (×50)	RMSE (×50)	MAPE	MAE (×50)	RMSE (×50)	MAPE	MAE (×50)	RMSE (×50)	MAPE
DCRNN (6.4)	2.221	2.365	5.61%	2.353	2.466	5.71%	2.477	2.504	7.42%
ST-GCN (6.4)	2.176	2.399	5.83%	2.318	2.398	6.10%	2.463	2.457	8.30%
Graph WaveNet (6.4)	2.199	2.309	5.48%	2.322	2.351	5.58%	2.456	2.438	7.22%
STTF (6.4)	2.105	2.237	5.45%	2.276	2.292	5.42%	2.429	2.425	7.13%
DCRNN (6.9)	2.372	2.448	5.75%	2.439	2.527	5.76%	2.555	2.53	7.48%
ST-GCN (6.9)	2.392	2.505	5.86%	2.462	2.534	6.27%	2.507	2.611	8.40%
Graph WaveNet (6.9)	2.301	2.338	5.54%	2.513	2.464	5.59%	2.641	2.509	7.29%
STTF (6.9)	2.204	2.423	5.49%	2.313	2.413	5.53%	2.524	2.474	7.26%

ability of the model for different time periods of the day. With reference to the peak traffic periods, we divide the day into three periods on average (04:00–12:00, 12:00–20:00, 20:00–24:00–04:00). Considering that the traffic changes are more significant during peak and off-peak periods in urban weekdays, the FUZHOU dataset (Monday, June 4, 2018) is chosen here for the experiments. Because graph-based deep learning models have a notable predictive advantage, we only use DCRNN, ST-GCN and Graph WaveNet to compare with our STTF model. From the results in Table 3, it is obvious that the STTF model has the advantageous and comprehensive performance, especially for the 30-minute prediction range.

Secondly, we take the large differences in traffic patterns between weekdays and weekends into account. For example, people wake up relatively later on weekends, so the morning peak is later and has a smaller peak. More people may

Table 5 Performance of STTF model with different time step

	Standard time steps	Metrics	
		STTF	
FUZHOU	5 min	MAE (×50)	2.497
		RMSE (×50)	2.505
	10 min	MAE (×50)	2.584
		RMSE (×50)	2.640
	15 min	MAE (×50)	2.696
		RMSE (×50)	2.731
PeMS-Bay	5 min	MAE (×50)	0.529
		RMSE (×50)	0.577
	10 min	MAE (×50)	0.603
		RMSE (×50)	0.689
	15 min	MAE (×50)	0.672
		RMSE (×50)	0.737

have time to go out and relax in the weekends' evening, so the roads may be more congested at night. Therefore, we re-select weekday hours (June 4, 2018, Monday, 00:00–24:00) and weekend hours (June 9, 2018, Saturday, 00:00–24:00) for the FUZHOU dataset to compare the generalization ability of the four graph-based learning models, respectively. From the results in Table 4, it is noticeable that the accuracy of all models for predicting weekend data is not significantly different from that of weekdays, and the STTF model has a marked predictive advantage.

Also, we need to consider the robustness of the model, i.e., to examine whether the model still maintains a comparable accuracy when the standard time step changes.

Therefore, for the specific time of the FUZHOU dataset (Monday, June 4, 2018, 6:00–20:00) and the specific time of the PeMS-Bay dataset (Monday, March 6, 2017, 6:00–20:00), we use different standard time steps (5 min, 10 min, and 15 min) to predict the *TCI* value using STTF model with a prediction range of 60 min and calculate MAE and RMSE, as shown in Table 5. Comparing Table 5 with Tables 1 and 2, it can be seen that the prediction accuracy of the STTF model decreases as the step length becomes longer, although it still has a competitive prediction ability under the variation of the standard step length. This is also due to its reduced amount of learning for temporal information. In the case of longer standard step

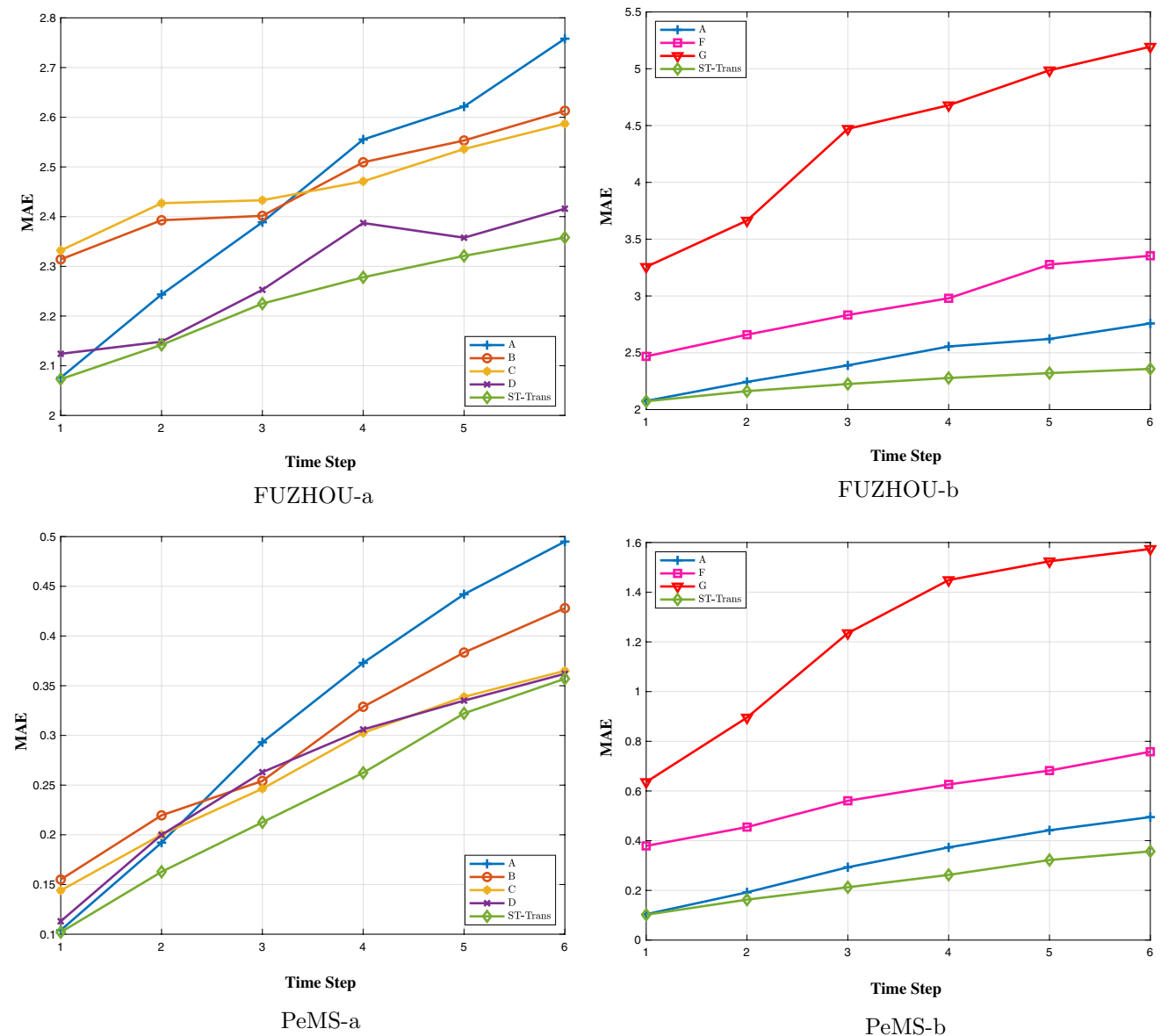


Fig. 8 Ablation experiments (display with MAE×50)

Table 6 Training and predicting efficiency of each model

Modal	Training time (s/ epoch)	Inference time (s)
ARIMA	3.94	28.84
PrePCT	698.04	209.43
DCRNN	834.89	137.93
ST-GCN	78.76	102.50
Graph WaveNet	219.98	9.37
STTF	243.93	12.52

length, the prediction range remains the same, resulting in a reduction in the amount of input temporal information and thus a decrease in the model's learning ability for temporal feature information.

Considering the many improvements of STTF over the classical Transformer structure, the ablation experiment is introduced here to examine the contribution of each structure in the STTF model. We predict the 30-min *TCI* values for the FUZHOU and PeMS-Bay datasets at the specific time given above, with a standard time step of 5 min. Among them, experiment (a) removes Spatial Att layer (A), Temporal Att layer (B), Masked-Temporal Att layer (C) and E-D Att layer (D), respectively, and experiment (b) removes the Spatial Att layer, Spatial Att layer & Temporal Att layer (F), Spatial Att layer & Temporal Att layer & Masked-Temporal Att layer (G), respectively. Finally, we calculate the MAE of the predicted values for each time step. The results are shown in Fig. 8. We can see that STTF consistently outperforms the model with the remaining incomplete models, indicating the ability of the four modules to mine spatio-temporal information. In particular, it should be noted that the missing Spatial Att layer module causes a particularly significant decrease in accuracy, especially in the long-term prediction range, which further indicates the long-term impact of the spatial structure of road networks on traffic congestion.

Finally, we compare the training time and prediction time of all six models on the PeMS-Bay dataset for the same time period (Monday, March 6, 2017, 6:00–20:00), the same standard time step (standard step size of 5 min), and the same prediction range (15 min), as shown in Table 6. As can be seen from the results, in the training phase, since ARIMA is a simple linear operation, its training time is absolutely superior. The last five algorithms are neural network algorithms, so there is a significant increase in the training time. Among them, ST-GCN is relatively efficient, but its prediction accuracy is far inferior to other graph neural networks. Graph WaveNet and STTF are the two models with the best and most similar overall performance. However, referring to the performance of the two models in Tables 1 and 2, we can see that STTF performs better.

6 Conclusion

We propose a new traffic congestion index and devise a STTF model based on data spatio-temporal information for predicting congestion on the road network. Specifically, we devise a new information embedding learning module that transforms both road network structure information and temporal information into feature vectors that can be learned by the network. The embedding vectors are learned by a new spatial attention module and a temporal attention module with different learning directions. The model has better prediction accuracy and relatively high efficiency compared with the state-of-the-art algorithm under real-world data.

Acknowledgements The authors would like to thank the anonymous reviewers for providing helpful comments.

Author Contributions RZ and XW completed the writing of the thesis, FZ conducted the guidance of the thesis, LL conducted part of the experiment and grammar modification of the thesis, and FH conducted part of the experiment of the thesis.

Funding This research was funded by the Natural Science Foundation of China (Grant No. 61962038), the Foreign Cooperation Project of Fujian Provincial Department of Science and Technology (Grant No. 2020I0014), and in part by the Guangxi Bagui Teams for Innovation and Research (Grant No. 201979).

Availability of Data and Materials The FUZHOU datasets analyzed during the current study are not publicly available due to the confidentiality of private data of citizens. The PeMS-Bay datasets we used is publicly available at <https://github.com/SANDAG/PeMS-Datasets>.

Declarations

Conflict of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent to Publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Akhtar, M., Moridpour, S.: A review of traffic congestion prediction using artificial intelligence. *J. Adv. Transport.* **2021**, 8878011 (2021)
2. Lee, J., Hong, B., Lee, K., Jang, Y.J.: A prediction model of traffic congestion using weather data. In: 2015 IEEE International Conference on Data Science and Data Intensive Systems. IEEE, pp. 81–88 (2015)
3. Fouladgar, M., Parchami, M., Elmasri, R., Ghaderi, A.: Scalable deep traffic flow neural networks for urban traffic congestion prediction. In: 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 2251–2258 (2017)
4. Jain, S., Jain, S.S., Jain, G.: Traffic congestion modelling based on origin and destination. *Procedia Eng.* **187**, 442–450 (2017)
5. Adetiloye, T., Awasthi, A.: Multimodal big data fusion for traffic congestion prediction. In: *Multimodal Analytics for Next-Generation Big Data Technologies and Applications*. Springer, Cham, pp. 319–335 (2019)
6. Zhang, P., Qian, Z.S.: User-centric interdependent urban systems: using time-of-day electricity usage data to predict morning roadway congestion. *Transport. Res. Part C: Emerg. Technol.* **92**, 392–411 (2018)
7. Boarnet, M.G., Kim, E.J., Parkany, E.: Measuring traffic congestion. *Transp. Res. Rec.* **1634**(1), 93–99 (1998)
8. Lee, J., Hong, B.: Congestion score computation of big traffic data. In 2014 IEEE Fourth International Conference on Big Data and Cloud Computing. IEEE, pp. 189–196 (2014)
9. Nicholson, H., Swann, C.D.: The prediction of traffic flow volumes based on spectral analysis. *Transp. Res.* **8**(6), 533–538 (1974)
10. Yang, X., Luo, S., Gao, K., Qiao, T., Chen, X.: Application of data science technologies in intelligent prediction of traffic congestion. *J. Adv. Transp.* **2019**, 2915369 (2019)
11. He, F., Yan, X., Liu, Y., Ma, L.: A traffic congestion assessment method for urban road networks based on speed performance index. *Procedia Eng.* **137**, 425–433 (2016)
12. Alghamdi, T., Elgazzar, K., Bayoumi, M., Sharaf, T., Shah, S.: Forecasting traffic congestion using ARIMA modeling. In 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC). IEEE, pp. 1227–1232 (2019)
13. Wang, H., Liu, L., Dong, S., Qian, Z., Wei, H.: A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD-ARIMA framework. *Transportmetrica B: Transp. Dyn.* **4**(3), 159–186 (2016)
14. Zaki, J.F., Ali-Eldin, A.M., Hussein, S.E., Saraya, S.F., Areed, F.F.: Time aware hybrid hidden Markov models for traffic congestion prediction. *Int. J. Elect. Eng. Inform.* **11**(1), 1–17 (2019)
15. Zaki, J.F., Ali-Eldin, A., Hussein, S.E., Saraya, S.F., Areed, F.F.: Traffic congestion prediction based on Hidden Markov Models and contrast measure. *Ain Shams Eng. J.* **11**(3), 535–551 (2020)
16. Wen, F., Zhang, G., Sun, L., Wang, X., Xu, X.: A hybrid temporal association rules mining method for traffic congestion prediction. *Comput. Ind. Eng.* **130**, 779–787 (2019)
17. Feng, X., Ling, X., Zheng, H., Chen, Z., Xu, Y.: Adaptive multi-kernel SVM with spatial-temporal correlation for short-term traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **20**(6), 2001–2013 (2018)
18. Xing, Y.M., Ban, X.J., Liu, R.: A short-term traffic flow prediction method based on kernel extreme learning machine. In: 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, pp. 533–536 (2018)
19. Xing, Y., Ban, X., Liu, X., Shen, Q.: Large-scale traffic congestion prediction based on the symmetric extreme learning machine cluster fast learning method. *Symmetry* **11**(6), 730 (2019)
20. Alajali, W., Zhou, W., Wen, S., Wang, Y.: Intersection traffic prediction using decision tree models. *Symmetry* **10**(9), 386 (2018)
21. Liu, Y., Wu, H.: Prediction of road traffic congestion based on random forest. In: 2017 10th International Symposium on Computational Intelligence and Design (ISCID), Vol. 2. IEEE, pp. 361–364 (2017)
22. Wang, S., Huang, W., Lo, H.K.: Traffic parameters estimation for signalized intersections based on combined shockwave analysis and Bayesian Network. *Transp. Res. Part C: Emerg. Technol.* **104**, 22–37 (2019)
23. Sun, S., Chen, J., Sun, J.: Traffic congestion prediction based on GPS trajectory data. *Int. J. Distrib. Sens. Netw.* **15**(5), 1550147719847440 (2019)
24. Zhu, L., Krishnan, R., Guo, F., Polak, J.W., Sivakumar, A.: Early identification of recurrent congestion in heterogeneous urban traffic. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, pp. 4392–4397 (2019)
25. Chen, M., Yu, X., Liu, Y.: PCNN: Deep convolutional networks for short-term traffic congestion prediction. *IEEE Trans. Intell. Transp. Syst.* **19**(11), 3550–3559 (2018)
26. Zhang, Z., Jiao, X.: A deep network with analogous self-attention for short-term traffic flow prediction. *IET Intel. Transport Syst.* **15**(7), 902–915 (2021)
27. Tu, Y., Lin, S., Qiao, J., Liu, B.: Deep traffic congestion prediction model based on road segment grouping. *Appl. Intell.* **51**(11), 8519–8541 (2021)
28. Zhang, W., Yu, Y., Qi, Y., Shu, F., Wang, Y.: Short-term traffic flow prediction based on Spatio-temporal analysis and CNN deep learning. *Transportmetrica A: Transp. Sci.* **15**(2), 1688–1711 (2019)
29. Chen, Y.Y., Lv, Y., Li, Z., Wang, F.Y.: Long short-term memory model for traffic congestion prediction with online open data. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 132–137 (2016)
30. Bai, M., Lin, Y., Ma, M., Wang, P., Duan, L.: PrePCT: traffic congestion prediction in smart cities with relative position congestion tensor. *Neurocomputing* **444**, 147–157 (2021)
31. Ranjan, N., Bhandari, S., Zhao, H.P., Kim, H., Khan, P.: City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN. *IEEE Access* **8**, 81606–81620 (2020)
32. Bogaerts, T., Masegosa, A.D., Angarita-Zapata, J.S., Onieva, E., Hellinckx, P.: A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. Part C Emerg. Technol.* **112**, 62–77 (2020)
33. Di, X., Xiao, Y., Zhu, C., Deng, Y., Zhao, Q., Rao, W.: Traffic congestion prediction by spatiotemporal propagation patterns. In 2019 20th IEEE International Conference on Mobile Data Management (MDM). IEEE, pp. 298–303 (2019)
34. Huang, Z., Xia, J., Li, F., Li, Z., Li, Q.: A peak traffic congestion prediction method based on bus driving time. *Entropy* **21**(7), 709 (2019)
35. Dai, R., Xu, S., Gu, Q., Ji, C., Liu, K.: 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 (2020)
36. Elfar, A., Talebpour, A., Mahmassani, H.S.: Machine learning approach to short-term traffic congestion prediction in a connected environment. *Transp. Res. Rec.* **2672**(45), 185–195 (2018)
37. Pan, Z., Zhang, W., Liang, Y., Zhang, W., Yu, Y., Zhang, J., Zheng, Y.: Spatio-temporal meta learning for urban traffic prediction. *IEEE Trans. Knowl. Data Eng.* (2020)
38. Park, C., Lee, C., Bahng, H., Tae, Y., Jin, S., Kim, K., Ko, S., Choo, J.: ST-GRAT: a novel Spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed.

- In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1215–1224 (2020)
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (2017)
 40. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J.: Transformers: state-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45 (2020)
 41. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
 42. Lim, B., Arik, S.Ö., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **37**(4), 1748–1764 (2021)
 43. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, pp. 1067–1077 (2015)
 44. Rodriguez, P., Bautista, M.A., Gonzalez, J., Escalera, S.: Beyond one-hot encoding: lower dimensional target embedding. *Image Vis. Comput.* **75**, 21–31 (2018)
 45. Yuan, J., Zheng, Y., Zhang, C., Xie, X., Sun, G.Z.: An interactive-voting based map matching algorithm. In: 2010 Eleventh international conference on mobile data management. IEEE, pp. 43–52 (2010)
 46. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint [arXiv:1707.01926](https://arxiv.org/abs/1707.01926) (2017)
 47. Ma, J., Yarats, D.: On the adequacy of untuned warmup for adaptive optimization. arXiv preprint [arXiv:1910.04209](https://arxiv.org/abs/1910.04209) (2019)
 48. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, pp. 249–256 (2010)
 49. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint [arXiv:1709.04875](https://arxiv.org/abs/1709.04875) (2017)
 50. Wu, Z., Pan, S., Long, G., Jiang, J. and Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. (2019) arXiv preprint [arXiv:1906.00121](https://arxiv.org/abs/1906.00121)

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