Sensors and Materials, Vol. 36, No. 8 (2024) 3257–3273
MYU Tokyo

S & M 3731

DSTLNet: Dynamic Spatial-Temporal Correlation Learning Network for Traffic Sensor Signal Prediction

Yuxiang Shan, Hailiang Lu, and Weidong Lou*

China Tobacco Zhejiang Industrial Company Limited, Hangzhou 311500, China

(Received December 19, 2023; accepted February 5, 2024)

Keywords: traffic prediction, artificial intelligence, path planning, graph convolutional network

Intelligent transportation systems based on sensor signals are crucial in addressing contemporary transportation issues, accomplishing dynamic traffic management, and facilitating route planning. However, the highly dynamic and intricate nature of traffic sensor signals presents difficulties for traffic prediction, with current models for traffic prediction inadequate in meeting the requirements of both long-term and short-term prediction tasks. In this paper, we propose a novel deep-learning framework called dynamic spatial-temporal correlation learning network (DSTLNet) that jointly leverages dynamical spatial and temporal features of traffic sensor signals to further improve the accuracy of long- and short-term traffic modeling and route planning. Specifically, we leverage the temporal convolutional network to capture long-term correlations. In addition, a spatial graph convolutional network is developed to dynamically model spatial features, and long- and short-term fusion layers are used to fuse the extracted long- and short-term temporal features, respectively. Experimental results on real-world datasets show that DSTLNet is competitive with the state-of-the-art, especially for long-term traffic prediction.

1. Introduction

With the advancement of embedded and intelligent technologies, there has been growing interest in intelligent transportation systems designed to effectively manage urban traffic and navigation. Their key components, namely, traffic signals, traffic modeling, and navigation, play an increasingly important role in traffic management. Previous studies have focused on developing efficient methods based on traffic sensor signals for traffic modeling and route planning. Traffic modeling and route planning require accurate predictions of future traffic conditions (such as traffic volume, speed, and density) within a specific time frame that are based on historical traffic observations. By accurately and efficiently predicting traffic conditions in advance, issues such as traffic congestion and accidents can be better addressed. However, owing to the complex spatial and temporal characteristics of traffic-sensing signals, precise traffic modeling and route planning remain challenging tasks in navigation.

*Corresponding author: e-mail: louweidong66@outlook.com
https://doi.org/10.18494/SAM4814

ISSN 0914-4935

In recent years, many efficient approaches have been proposed for traffic modeling and forecasting.[1,2] Forecasting tasks in traffic can be classified into short-term (5–30 min) and long-term (30–60 min), depending on the length of the forecasting time interval. To efficiently extract temporal and spatial correlations, key techniques are required in traffic modeling and forecasting.

The current methods for traffic modeling and forecasting can be broadly classified into two categories: statistical methods and deep-learning-based methods. Although conventional methods such as Kalman filtering[3] have exhibited good prediction performance, they make static assumptions that limit their performance in solving long-term forecasting tasks, where traffic observations vary over a long-term interval. On the other hand, neural networks have shown potential for capturing the temporal and spatial structures of training data.

As deep-learning techniques have developed, some deep-learning-based traffic modeling and forecasting approaches have been presented. Among them, recurrent neural networks (RNNs) are widely used for modeling temporal features.[4–7] However, RNNs and their variants such as long short-term memory (LSTM) and gated recurrent units (GRUs) are usually based on sequential processing; thus, they only remember the latest information and cannot adapt to solve long-term sequences. Convolutional neural networks (CNNs) are widely used to model spatial features,[8] although they have unsatisfactory performance when solving non-Euclidean spatial features. By integrating RNNs and CNNs, some studies have extracted spatial and temporal features simultaneously and achieved excellent modeling and forecasting performance.[9]

In recent studies, traffic prediction solutions have explored graph learning extensively to model the time-varying traffic topology.[10,11] For example, the dynamic multi-faceted spatial-temporal graph convolution network (DMSTGCN)[12] constructs a tensor to capture spatial correlations by studying the dynamic graph structure at each time. Additionally, it incorporates primary and auxiliary feature extraction structures. The multi-task graph neural network (MTGNN)[13] utilizes a graph learning and inception structure to improve the accuracy of modeling and prediction. It also uses mask techniques to aggregate the graph structure. However, existing traffic prediction methods face challenges in solving both short-term and long-term modeling and forecasting simultaneously. Moreover, the spatial and temporal correlations in the traffic network are difficult to capture together. To address these challenges, in this paper, we propose a novel deep-learning framework that focuses on short-term and long-term traffic modeling and forecasting by capturing varying spatial-temporal correlations simultaneously.

To address the aforementioned challenges, this paper focuses on the issue of predicting short- and long-term traffic patterns. It proposes a novel deep-learning framework called the dynamic spatial-temporal correlation learning network (DSTLNet) to forecast future traffic conditions within specific time intervals. DSTLNet effectively captures the spatial properties and both short- and long-term temporal attributes of traffic data. The system initially splits the input traffic sequence into two categories, namely, long-term and short-term sequences, and processes each category independently. DSTLNet utilizes a temporal convolutional network (TCN) to handle the long-term sequence and subsequently encodes both short- and long-term temporal correlations through recurrent networks in the encoder. Then, a spatial graph convolutional network (GCN) is employed to capture sophisticated spatial relationships. Finally, the spatial and

temporal features are extracted and integrated into the prediction layer for forecasting future traffic conditions. This integration of AI and robotics not only enhances the accuracy of traffic modeling and forecasting but also significantly contributes to the development of intelligent and efficient traffic management systems. The key findings and contributions outlined in this paper can be shown as follows.

(1) A long short-term temporal processing module is proposed to model periodic dependences of data. In particular, the module contains a TCN and a long short-term fusion component. The TCN is used to extract periodic features from long-term data. The long short-term fusion component is used to generate attention scores for both long- and short-term data, enabling the model to learn long-term and short-term dependences of the data, respectively.
(2) A spatial graph convolution module is presented to adaptively model the time-varying spatial correlations among nodes that demonstrate comparable patterns in geographic locations with a flexible adjacency matrix.
(3) The efficacy of the proposed approach was tested on real traffic datasets, with experimental outcomes indicating that it outperforms the most sophisticated techniques.

The rest of the paper is organized as follows. Section 2 briefly reviews existing approaches for traffic prediction. Section 3 develops some definitions of traffic prediction and formalizes the problem. Section 4 elaborates on the proposed DSTLNet model. Section 5 reports extensive experiments conducted on real-world traffic datasets to evaluate DSTLNet. Finally, we conclude this paper and discuss further work in Sect. 6.

2. Related Work

This section provides an overview of existing studies in the field of traffic prediction. Considering the components of our proposed model, we review related works from two aspects: GCN-based traffic prediction and attention-mechanism-based traffic prediction. We conclude by highlighting the distinctions between our study and existing surveys.

2.1 GCN-based traffic prediction

Numerous studies have considered a traffic network as a graph structure and implemented a GCN to leverage the spatial correlations. Many GCN-based traffic prediction models have been proposed and have obtained excellent prediction results. In particular, Chen *et al.*[14] designed an LSTM network based on a stacked structure, in which a GCN was used to obtain multiple features to realize model complex spatial dependences. Zhao *et al.*[5] proposed the integration of a GRU and a GCN model for traffic prediction. Zhang *et al.*[15] employed a GCN module to model the dynamic spatial dependences among traffic segments. Li *et al.*[6] utilized a random walk strategy to model spatial correlations of traffic networks. Zhang *et al.*[16] combined a GCN module and a feedforward neural network to realize traffic forecasting. Wu *et al.*[17] integrated a GCN and a TCN to solve traffic forecasting tasks.

2.2 Attention-mechanism-based traffic prediction

The attention mechanism is efficient and effective for time series modeling and prediction, as it can focus on important features to enhance accuracy. For example, Zeng *et al.*[18] implemented the attention mechanism to leverage spatial dependences in an entire traffic network. Similarly, Bai *et al.*[19] utilized the attention mechanism and a spatial-temporal dynamic network to extract time-varying correlations in a traffic system. Liang *et al.*[20] proposed a hierarchical multi-level attention-mechanism-based framework to model dynamic spatial and temporal correlations. Zhang *et al.*[21] approached the modeling problem by considering both spatial and temporal dimensions. Finally, Kong *et al.*[22] adopted a multi-head attention-mechanism-based model to capture both global and local spatial dependences.

In summary, a GCN, a TCN, and the attention mechanism are efficient in modeling spatial and temporal dependences for traffic prediction. Nonetheless, most existing models fail to perform well in both long- and short-term predictions. Therefore, in this study, we propose a novel framework for traffic prediction that addresses this issue. Our proposed model exhibits excellent performance, as demonstrated by experiments on real-world datasets.

3. Preliminaries

In this section, we begin by presenting the definitions utilized in this paper. Following this, we formalize the traffic flow prediction problem.

3.1 Definitions

Definition 1 (road network)

Given an undigraph $G = (V, E, A)$ as the road network, $V = \{v_i\}_{i=1,2,...,N}$ is the set of nodes, $E = e_{ij}$ is the set of edges, and $A \in R^{N\times N}$ is the adjacency matrix.

Definition 2 (traffic flow)

Given an undigraph $G = (V, E, A)$, the nodes represent the corresponding positions of the sensors in the road network. The traffic data collected from the road network are denoted as $X \in R^{N\times T}$. In particular, $X^{(t)}$ is the traffic data at the time step t. In this paper, we focus on traffic speed data without sacrificing any generality.

3.2 Problem formalization

Given the graph $G = (V, E, A)$ and T historical traffic observations $\{X^{(1)}, ..., X^{(T)}\} \in R^{N\times T}$, our goal is to learn a function $f(\bullet)$ that can estimate the most probable traffic conditions for the next time step $\{X^{(T+1)}, ..., X^{(T+H)}\} \in R^{N\times H}$:

$$\{X^{(1)},...,X^{(T)}\} \overset{f(\bullet)}{\rightarrow} \{X^{(t+1)},...,X^{(t+H)}\}, \tag{1}$$

4. Methodology

In this section, we introduce DSTLNet, a novel approach for modeling and forecasting traffic. The architecture of DSTLNet is displayed in Fig. 1 and comprises six components: a fully connected layer, a temporal convolution layer, long- and short-term fusion layers, a spatial-temporal convolution layer, and a prediction layer. We elaborate on the design of each component in the subsequent sections.

4.1 Input component

Most current research focuses on using the entire traffic sequence as the input to train models, which are then used to predict future traffic conditions across the network. However, traffic data for both short- and long-term periods may have multi-scale temporal features. As illustrated in Fig. 1, we designate the short- and long-term sequences, which are then integrated into the model. The lengths of the short- and long-term sequences are respectively denoted as L_s and L_l. Specifically, the short-term sequence is defined as

$$X_S = \{X^{(1,n)},...,X^{(Ts-1,n)},X^{(Ts,n)}\}^T, \tag{2}$$

where $X_S \in R^{nTs \times N}$, Ts is the number of time steps in the proposed model, and $X^{(Ts,n)}$ is the traffic data collected at Ts in the short-term period. The long-term sequence is defined as

$$X_L = \{X^{(1,1)},X^{(2,1)},...,X^{(Ts,1)},X^{(1,2)},...,X^{(Ts,n)}\}^T, \tag{3}$$

where $X_L \in R^{nTs \times N}$, $X^{(Ts,N)}$ is the traffic data collected at Ts in the nth period. For simplicity, we use Tp to denote the number of the period.

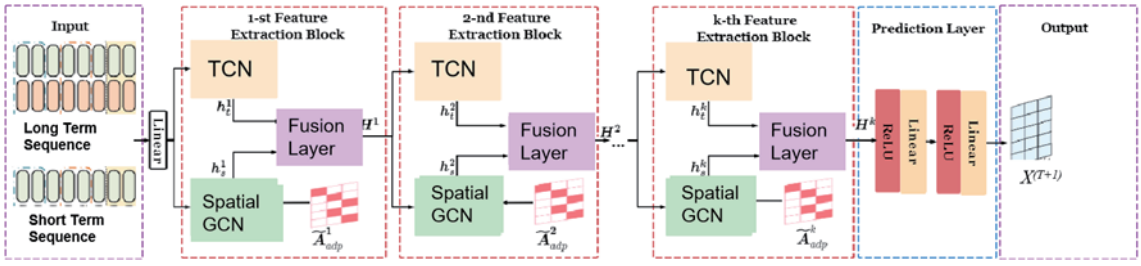


Fig. 1. (Color online) Architecture of DSTLNet. Its structure comprises stacked feature extraction blocks and a prediction layer. Each extraction block incorporates a TCN and a spatial GCN, which work together to capture both temporal and spatial correlations. To incorporate multi-scale spatial and temporal features, long- and short-term fusion is implemented.

4.2 Fully connected layer

The long-term sequence X_L and short-term sequence X_S are first fed into the fully connected layer to exploit the features of the input sequence. The operations are formulated as

$$\begin{aligned} Z_S &= f_c(X_S) = ReLU\left(X_S w_1 + b_1\right), \\ Z_L &= f_c(X_L) = ReLU(X_L w_2 + b_2), \end{aligned} \tag{4}$$

where w_1, w_2, b_1, and b_2 are learnable parameters, and Z_S and Z_L are the outputs of the fully connected layer. Z_L is then fed into the next temporal convolution layer to extract the long-term temporal correlations.

4.3 Temporal convolution for long-term temporal correlations

Traffic conditions during an interval are correlated with historical observations; thus, it is beneficial to incorporate temporal correlations into traffic forecasting models. Recent studies have presented the advantage of TCNs in processing time series data, We therefore utilize a TCN to learn the long-term temporal correlations. Inspired by Wu *et al.*,[17] we design the temporal convolution layer using stacking dilated convolution layers with progressively larger dilation factors, which can capture the multi-scale temporal correlations and obtain the receptive field. Moreover, to adaptively adjust the flows from different dilated convolution layers, we add a gating mechanism. The following is the definition:

$$H_{LT} = \tanh(W_1 * Z_L) \odot \sigma(W_2 * Z_L), \tag{5}$$

where w_1 and w_2 are learnable parameters. $\odot$ is an elementwise multiplication operator, $\sigma(\bullet)$ denotes the *sigmoid* function, * is used to denote the dilated convolution operation, and H_{LT} is the output of the temporal convolution layer.

Given an input sequence Z_L, the dilated convolution operation is defined as

$$Z_L * f_{1\times k}\left(s\right) = \sum_{i=0} f_{1\times k}\left(i\right) Z_L\left(s - d \times i\right), \tag{6}$$

where d is the dilation factor, k is the convolution kernel size, and $f_{1\times k}$. denotes the 1D convolution operator. When the network requires a large receptive field, the dilated convolution is efficient owing to its reduced computational burden.

4.4 Long- and short-term fusion layers

The long- and short-term fusion layers are responsible for further extracting multi-scale temporal features by integrating the output of the temporal convolution layer H_{LT} and the short-

term sequence Z_S. These branches operate in parallel and capture temporal dependences at different scales. The fusion process is formulated as follows.

First, the short-term sequence Z_S is processed through two convolution operations α_1 and α_2, followed by a softmax operation. The result is multiplied elementwise with the original short-term sequence Z_S and passed through a nonlinear activation function F. This process is expressed as

$$s^k = F\left[\alpha_1\left(Z_S\right) \times softmax\left(\alpha_2\left(Z_S\right)\right)\right]. \tag{7}$$

Similarly, the long-term representation H_{LT} is processed through two convolution operations, λ_1 and λ_2. The results of these convolutions are passed through a softmax operation and multiplied elementwise. The resulting vector is then passed through the nonlinear activation function F. This process is expressed as

$$l^k = F\left[\left(softmax\left(2\lambda_1\left(H_{LT}\right)\right)\right) \times \left(2\lambda_2\left(H_{LT}\right)\right)\right]. \tag{8}$$

Next, the obtained weight vectors s^k and l^k are multiplied elementwise with the short-term sequence Z_S and the long-term representation H_{LT}, respectively. The operator $\odot^s$ denotes a channelwise multiplication and $\odot^l$ denotes a spatialwise multiplication. This fusion operation is expressed as.

$$H_F = s^k \odot^s Z_{\mathbf{S}} + l^k \odot^l H_{LT}. \tag{9}$$

Specifically, the channelwise multiplication operator is applied to the short-term sequence Z_S and the weight vector s^k. It performs an elementwise multiplication between the corresponding elements of the two vectors across the channel dimension. The purpose of this operation is to assign different weights to different channels of the short-term sequence, enabling selective emphasis on specific features or channels based on their relevance or importance. By multiplying the weight vector with the short-term sequence, the fusion process can enhance channel-specific information, allowing the model to focus on the most relevant information for the fusion result. Moreover, the spatialwise multiplication operator is applied to the long-term representation H_{LT} and the weight vector l^k. It performs an elementwise multiplication between the corresponding elements of the two vectors across the spatial dimensions. The purpose of this operation is to assign different weights to different spatial locations or positions in the long-term representation. This allows the fusion process to selectively emphasize the contribution of specific spatial locations or positions in the long-term representation. By multiplying the weight vector with the long-term representation, the fusion process can de-emphasize certain spatial information, enabling the model to focus on the most relevant spatial aspects for the fusion result .

4.5 Spatial-temporal convolution module

Given graph $G = (V, E, A)$ and node $v_i \in V$ on G, the correlations among v_i and its neighbor nodes are described using an adjacency matrix. The spatial-temporal convolution module is formulated as

$$H_S = AH_T W, \tag{10}$$

where W is the model parameter matrix.[23]

The diffusion convolution has been widely used in the modeling of spatial and temporal characteristics.[6] According to the model, the diffusion process of graph signals is treated as K finite steps. Inspired by Wu *et al.*,[17] we thus formulate the diffusion convolution as

$$H_S = \sum_{k=0} P^k H_T W_k, \tag{11}$$

where $P^k = A/rowsum(A)$. The output H_S is then fed into the prediction layer to obtain the final forecasting results.

4.6 Prediction layer and loss function

The extracted temporal and spatial correlations are integrated into the prediction layer Fp to acquire the final forecasting results, which is presented as

$$Fp(H_S) = \hat{Y}, \tag{12}$$

Specially, the operations of the prediction layer are defined as

$$\begin{aligned} Y' &= W_1(\sigma(G^{HS})) + b_1, \\ \hat{Y} &= W_2(\sigma(Y')) + b_2, \end{aligned} \tag{13}$$

where W_1, W_2, b_1, and b_2 are learnable parameters and σ is the activation function. To train DSTLNet to forecast the future traffic conditions $\hat{Y}$ given the historical traffic conditions Y, we choose the mean absolute error (MAE) as the loss function, which is defined as

$$L(\theta) = 1/n \sum_{i=0} | Y - \hat{Y} |, \tag{14}$$

where θ denotes the learnable parameters in the proposed model.

5. Experiments

In this section, we report the empirical assessment of DSTLNet and its competitive baselines on two publicly accessible traffic datasets. The essential concept behind DSTLNet is to acquire spatial and temporal traits from traffic observations by utilizing the extracted spatial-temporal correlations to forecast future traffic conditions. We present experimental findings to substantiate the efficiency of our proposed model.

5.1 Experimental setup

5.1.1 Dataset description

Our experiments utilize authentic traffic datasets, specifically the publicly available METR-LA and PeMS-BAY datasets, which were collected from loop detectors located on highways in Los Angeles and California, respectively.[6] For more information on the datasets, refer to Table 1.

- **METR-LA**: Traffic data collected between January 1 and May 31, 2017 by 207 loop detectors located in Los Angeles County.
- **PeMS-BAY**: Traffic data collected between January 1 and May 31, 2017 by 325 loop detectors located in the San Francisco Bay Area.

5.1.2 Data preprocessing

Following previous studies, such as the one on Graph WaveNet,[17] the standard time intervals are set as 5 min. The datasets are partitioned into three components, consisting of 70% of the data for training the model, 20% for testing the model, and 10% for validating the model. Given that the datasets may have incomplete information, we employ linear interpolation to replace the missing values and use the Z-score method to standardize the input data. For the two datasets, the weighted adjacency matrix W is constructed as follows:

$$w_{ij} = \begin{cases} \exp(-\frac{d_{ij}^2}{\eta^2}), & i \neq j \text{ and } \exp(-\frac{d_{ij}^2}{\eta^2}) \geq \varepsilon \\ 0 \quad \text{otherwise} \end{cases} \tag{15}$$

where w_{ij} represents the weight of the edge between the ith and jth road nodes, d_{ij} represents the distance between the ith and jth road nodes, and η and ε are parameters that control the distribution and sparsity of W.

Table 1
Details of the datasets used.

Dataset	Sensors	Time horizon	Time interval	Daily range
METR-LA	207	34272	5 min	00:00–24:00
PeMS-BAY	325	52116		

5.2 Evaluation metrics

To assess the prediction performance of various methods, we employed the following evaluation metrics to compare the discrepancy between the actual value Y_t and the predicted outcome $\hat{Y}_t$:

1) Mean absolute error:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_t - \hat{Y}_t\right|, \tag{16}$$

2) Mean absolute percentage error (*MAPE*):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{\left|Y_t\right|}, \tag{17}$$

3) Root mean squared error (*RMSE*):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_t - \hat{Y}_t\right)^2}. \tag{18}$$

5.3 Parameter settings

We conducted all our experiments on a workstation equipped with an Intel(R) Core(TM) I7-11700K CPU@2.50 GHz processor and an NVIDIA GeForce RTX 3060 system. To predict traffic flows in the next 15, 30, and 60 min (3, 6, and 12 time series, respectively), we utilized historical observations and optimized our model's parameters on the validation set. Our best performance was achieved for $K = 7$. During the training phase, we used a batch size of 64 and an initial learning rate of 0.001, and we implemented dropout on the graph convolution layer. Dropout involves temporarily discarding neural network units from the network according to a certain probability ($P = 0.3$ in our case) during the training process of a deep-learning network. We set T_S to 24 and the number of periods, n, to 7.

5.4 Baselines

The competing baseline approaches were as follows:

Conventional models:

- **HA**: Uses the average of the previous observations as final results.
- **ARIMA**[24]: Auto-regressive integrated moving average.

Deep neural network models:

- **FC-LSTM**[25]: Fully connected LSTM.
- **DCRNN**[6]: Diffusion convolutional RNN.
- **STGCN**[26]: Spatial-temporal GCN.
- **GMAN**[27]: Graph multi-attention network.
- **DMSTGCN**[12]: Dynamic multi-faceted spatial-temporal GCN.
- **Graph WaveNet**[17]: Uses GCN and dilated casual convolution to model spatial-temporal correlations.
- **S^{2}TAT**[28]: Employs the attention mechanism and a GCN module to extract spatial-temporal correlations.

5.5 Experimental results

5.5.1 Results

Table 2 presents a comparison between the performance characteristics of DSTLNet and the baseline models for predicting 15, 30, and 60 min intervals on the METR-LA and PeMS-BAY datasets. From this table, we conclude the following:

(1) The performance of conventional methods such as HA and ARIMA is unsatisfactory, implying that conventional methods have limited ability to model complex traffic data. In particular, DSTLNet outperforms ARIMA in capturing complex temporal patterns and handling long-term dependences. ARIMA relies on linear models and is limited in capturing nonlinear and spatial dependences present in traffic data, which DSTLNet effectively addresses.

(2) Methods that consider only temporal features, including FC-LSTM and WaveNet, can achieve better results for short-term prediction tasks than for long-term prediction tasks. It was also observed that with increasing time interval, their prediction accuracy markedly decreases. Compared with the above methods, the methods that capture spatial-temporal correlations perform better, such as DCRNN, STGCN, GMAN, DMSTGCN, and Graph WaveNet. In particular, DSTLNet offers advantages over LSTM networks by incorporating a long short-term fusion component. This allows DSTLNet to explicitly model both long-term and short-term dependences, whereas LSTM networks primarily focus on capturing long-term dependences. Although CNNs can capture spatial patterns, DSTLNet surpasses them by incorporating a spatial graph convolution module. This module adaptively models time-varying spatial correlations among nodes, allowing DSTLNet to capture dynamic spatial relationships in traffic data more effectively.

(3) DSTLNet outperforms the other baseline models in terms of prediction accuracy for both long-term and short-term predictions, with markedly improved performance on both datasets for all time periods, although the prediction accuracy decreased with increasing time period. We thus conclude that the framework based on complex spatial-temporal correlation modeling has excellent performance in processing spatial-temporal data. The most important contributions of DSTLNet lie in solving both long- and short-term traffic predictions. For

Table 2
Performance comparison of different approaches on the METR-LA and PeMS-BAY datasets.

Model	METR-LA (15/30/60 min)		
	MAE	*MAPE* (%)	*RMSE*
HA	4.16/4.16/4.16	13.00/13.00/13.00	7.80/7.80/7.80
ARIMA	3.99/5.15/6.90	9.60/12.70/17.40	8.21/10.45/13.23
FC-LSTM	3.44/3.77/4.37	9.60/10.90/13.20	6.30/7.23/8.69
DCRNN	2.77/3.15/3.60	7.30/8.80/10.50	5.38/6.45/7.60
STGCN	2.88/3.47/4.59	7.62/9.57/12.70	5.74/7.24/9.40
GMAN	4.04/4.59/5.33	10.26/11.69/13.60	8.53/9.85/11.21
DMSTGCN	2.85/3.26/3.72	7.54/9.19/10.96	5.54/6.56/7.55
Graph WaveNet	2.69/3.07/3.53	6.90/8.37/10.01	5.15/6.22/7.37
S2TAT	2.78/3.10/3.43	7.38/8.70/10.02	5.43/6.39/7.32
DSTLNet (ours)	2.65/2.98/3.40	6.83/8.34 /9.50	4.88/5.99/6.95

Model	PeMS-BAY (15/30/60 min)		
	MAE	*MAPE* (%)	*RMSE*
HA	2.88/2.88/2.88	6.77/6.77/6.77	5.59/5.59/5.59
ARIMA	1.62/2.33/3.38	3.50/5.40/8.30	3.30/4.76/6.50
FC-LSTM	2.05/2.20/2.37	4.80/5.20/5.70	4.19/4.55/4.96
DCRNN	1.38/1.74/2.07	2.90/3.90/4.90	2.95/3.97/4.74
STGCN	1.36/1.81/2.49	2.90/4.17/5.79	2.96/4.27/5.69
GMAN	1.34/1.62/1.86	2.81/3.63/4.31	2.82/3.72/4.32
DMSTGCN	1.33/1.67/1.99	2.80/3.81/4.78	2.83/3.79/4.54
Graph WaveNet	1.30/1.63/1.95	2.73/3.67/4.63	2.74/3.70/4.52
S2TAT	1.33/1.62/1.85	2.83/3.67/4.31	2.89/3.73/4.30
DSTLNet (ours)	1.20/1.52/1.82	2.64/3.08/3.95	2.65/3.45/4.12

long-term prediction, the TCN component in DSTLNet efficiently extracts periodic features from long-term traffic data. This enables DSTLNet to capture long-term trends, seasonality, and periodic patterns, leading to improved long-term traffic prediction accuracy compared with models that solely focus on short-term dependences. On the other hand, for short-term prediction, the long short-term fusion component in DSTLNet generates attention scores for both long- and short-term data. This allows the model to effectively learn short-term dependences, enabling accurate short-term traffic prediction. DSTLNet combines the long-term and short-term modeling strengths, resulting in more accurate predictions across various time horizons.

Furthermore, we analyze the impact of various factors on model performance as follows.

a. Input data feature: DSTLNet is flexible and can handle various input data characteristics, such as traffic volume, speed, and historical patterns. By leveraging both temporal and spatial information, DSTLNet can adapt to different data patterns and capture the underlying dynamics.
b. Model architecture variations: DSTLNet's modular architecture allows for flexibility in incorporating additional components or modifying existing ones. Experimenting with different architectures, such as varying the number of layers or the size of hidden units, can help optimize the model's performance based on specific traffic prediction tasks.
c. Training strategies: The performance of DSTLNet can be affected by training strategies, such as the choice of optimization algorithms, learning rate schedules, and regularization

techniques. Fine-tuning these strategies can enhance the model's generalization and improve its ability to capture temporal and spatial dependences in traffic data.

5.5.2 Ablation study

We also conducted an ablation study to explore the effects of different components on DSTLNet for the METR-LA dataset. The three key parts of the mode are the spatial-temporal graph convolution module, the temporal convolution for long-term correlation module, and the long- and short-term fusion layer module. On the basis of the model, we propose three variants: DSTLNet-GCN, DSTLNet-TCN, and DSTLNet-LSF.

- **DSTLNet-GCN**: DSTLNet without the spatial-temporal graph convolution module.
- **DSTLNet-TCN**: DSTLNet without temporal convolution for the long-term correlation module.
- **DSTLNet-LSF**: DSTLNet without the long- and short-term fusion layer module.

Table 3 presents the ablation results for the models, which indicate that the spatial-temporal graph convolution module has the most significant impact on the prediction results among the components. Removing this module increases *RMSE* from 9.5 to 10.03 for the METR-LA dataset, suggesting that the spatial-temporal graph convolution module plays an important role in improving the model performance. The temporal convolution has the second highest impact on the prediction results, validating its effectiveness in modeling temporal features. The long- and short-term fusion layer module also affect the performance, as *RMSE* increases from 9.5 to 9.61 when this module is removed.

5.5.3 Effect of number of feature extraction blocks, k

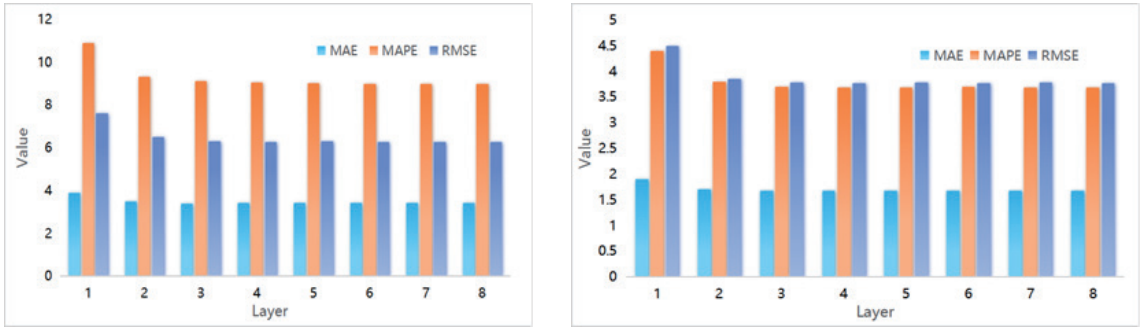
We next investigated the effect of the number of stacked feature extraction blocks, k, on the prediction accuracy. Figure 2 shows the results for the METR-LA and PeMS-BAY datasets. It can be clearly observed that k affects the prediction accuracy, with the best performance obtained for $K = 8$. Note that k can be adjusted to improve the traffic forecasting performance.

5.5.4 Effect of adaptive adjacency matrix

We also examined the impact of the learned adaptive adjacency matrix for the METR-LA dataset. The heat map in Fig. 3 illustrates the graph structure, revealing columns with high-value

Table 3
Results of ablation study.

Model	METR-LA (60 min)		
	MAE	*MAPE* (%)	*RMSE*
DSTLNet-GCN	3.54	10.03	7.26
DSTLNet-TCN	3.47	9.81	7.04
DSTLNet-LSF	3.42	9.61	6.98
DSTLNet (ours)	3.40	9.50	6.95



(a) (b)

Fig. 2. (Color online) Effect of k for the METR-LA and PeMS-BAY datasets. (a) METR-LA and (b) PeMS-BAY.

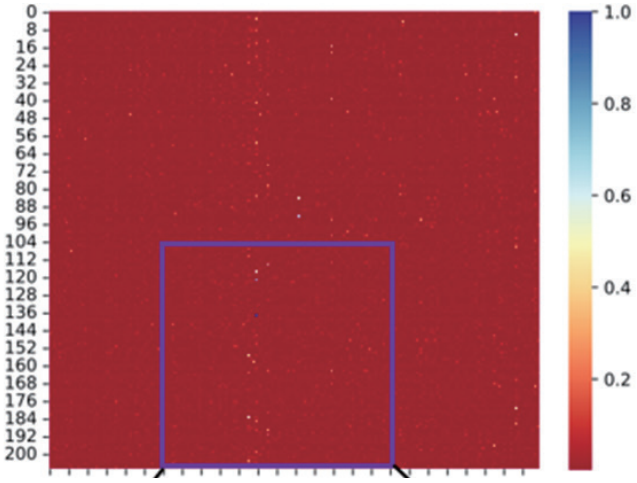


Fig. 3. (Color online) Heat map of the graph structure.

nodes that have a higher effect on the other nodes in the graph. This suggests that mutual effects exist among nodes. It is evident that the adaptive adjacency matrix helps to capture spatial correlations, leading to improved prediction results.

5.5.5 Computation time

Table 4 gives a comparison of the computational cost of different models for the METR-LA dataset. DSTLNet has a training speed about five times higher than that of DCRNN. It also has a higher speed than DMSTGCN and GMAN. STGCN outperforms all other models owing to its non-autoregressive nature. Overall, DSTLNet has the third lowest training time and the third lowest inference time. The table also indicates that the time cost of DSTLNet is competitive with that of Graph WaveNet. Although STGCN has the shortest training and inference times, it has low prediction performance. The model with the second-highest performance, Graph WaveNet (see Table 2), is outperformed by DSTLNet in both short- and long-term predictions.

When considering the trade-off between speed and accuracy in the prediction model, it is important to consider typically finding the right balance between the computational efficiency

Table 4
Time cost for METR-LA.

Dataset	Model	Computation time	
		Training (s/epoch)	Inference
METR-LA	DCRNN	230.31	15.73
	STGCN	32.8	7.37
	Graph WaveNet	51.3	1.84
	DMSTGCN	63.5	1.05
	GMAN	410.4	21.8
	DSTLNet	55.2	3.10

of the model and the quality of the prediction it generates. It is observed that Graph WaveNet and DMSTGCN are better than the proposed DSTLNet in terms of inference speed. The reason is that the proposed DSTLNet might prioritize accuracy by incorporating additional layers and components, such as long and short-term fusion components, which enhance the model's prediction capabilities for both long- and short-term predictions. Although this leads to improved prediction performance, it could result in slightly longer inference times compared to the faster models mentioned earlier.

6. Conclusions

In this paper, we have presented DSTLNet, a novel neural network framework that effectively addresses both long- and short-term traffic modeling and forecasting tasks. Our architecture incorporates temporal convolution to handle long-term traffic sequences, enabling the capture of multiple field features through dilated convolutions. Additionally, we have introduced long- and short-term fusion layers that combine temporal features from different time scales, along with a spatial graph convolution module that captures spatial characteristics using an adaptive adjacency matrix. The experimental results have demonstrated that DSTLNet performs comparably to existing baseline models when evaluated on public traffic datasets. This suggests the effectiveness of our proposed architecture in capturing both long- and short-term fluctuations in traffic conditions.

However, it is important to acknowledge the limitations of DSTLNet. One limitation is the current exclusion of external factors, such as traffic accidents and weather conditions, which can significantly impact traffic patterns. In future research, we will consider incorporating these external factors into the model to improve its prediction performance. Furthermore, while our framework has shown promising results in traffic modeling and forecasting tasks, we plan to explore extending DSTLNet to a wider range of spatial-temporal prediction tasks beyond traffic.

References

1 M. Xu, W. Dai, C. Liu, X. Gao, W. Lin, G. Qi, and H. Xiong: arXiv:2001.02908 (2020).
2 R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong: Proc. 29th Int. Joint Conf. Artificial Intelligence (2020) 2355–2361. https://doi.org/10.24963/ijcai.2020/326.https://doi.org/10.24963/ijcai.2020/326
3 I. Okutani and Y. J. Stephanedes: Transp. Res. Part B Methodol. **18** (1984) 1.

4 R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu: Proc. 2017 SIAM Int. Conf. Data Mining (2017) 777. https://doi.org/10.1137/1.9781611974973.87
5 L. Zhao, Y. Song, C. Zhang, Y. Liu, and H. Li: IEEE Trans. Intell. Transp. Syst. **21** (2020) 3848. https://doi.org/10.1109/TITS.2019.2935152
6 Y. Li, R. Yu, C. Shahabi, and Y. Liu: arXiv:1707.01926 (2017).
7 B. Du, H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu, and J. Li: IEEE Trans. Intell. Transp. Syst. **21** (2020) 972. https://doi.org/10.1109/TITS.2019.2900481
8 X. Ma, Z. Dai, Z. He, J. Ma, and Y. Wang: Sensors **17** (2017) 818. https://doi.org/10.3390/s17040818
9 Y. Wu and H. Tan: arXiv:1612.01022 (2016).
10 Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang: IEEE Trans. Intell. Transp. Syst. **22** (2021) 219. https://doi.org/10.1109/TITS.2019.2955794
11 S. Wang, M. Zhang, H. Miao, and P. S. Yu: Proc. 2021 SIAM Int. Conf. Data Mining (2021) 504–512. https://doi.org/10.1137/1.9781611976700.57. https://doi.org/10.1137/1.9781611976700.57
12 L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong: Proc. 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining (2021) 547–555. https://doi.org/10.1145/3447548.3467275
13 Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang: Proc. 26th ACM SIGKDD Conf. Knowledge Discovery & Data Mining (2020) 753–763. https://doi.org/10.1145/3394486.3403118
14 P. Chen, X. Fu, and X. Wang: IEEE Trans. Intell. Transp. Syst. **23** (2022) 6950.
15 M. Zhang, Y. Li, F. Sun, D. Guo, and P. Hui.: IEEE Int. Conf. Data Mining (2021) 1475–1480. https://doi.org/10.1109/ICDM51629.2021.00191
16 T. Zhang, W. Ding, T. Chen, Z. Wang, and J. Chen: Wirel. Commun. Mob. Comput. (2021) 1997212. https://doi.org/10.1155/2021/1997212
17 Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang: Proc. 28th Int. Joint Conf. Artificial Intelligence (2019) 1907–1913. https://doi.org/10.24963/ijcai.2019/264
18 H. Zeng, Z. Peng, X. Huang, Y. Yang, and R. Hu: Appl. Intell. **52** (2022) 10285. https://doi.org/10.1007/s10489-021-02879-1
19 J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li: ISPRS Int. J. Geo Inf. **10** (2021) 485. https://doi.org/10.3390/ijgi10070485
20 Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng: Proc. 27th Int. Joint Conf. Artificial Intelligence (2018) 3428. https://doi.org/10.24963/ijcai.2018/476
21 Y. Zhang, Y. Yang, W. Zhou, H. Wang, and X. Ouyang: Appl. Intell. **51** (2021) 6895. https://doi.org/10.1007/s10489-020-02074-8
22 X. Kong, J. Zhang, X. Wei, W. Xing, and W. Lu: Appl. Intell. **52** (2022) 4300. https://doi.org/10.1007/s10489-021-02648-0
23 T. N. Kipf and M. Welling: 5th Int. Conf. Learning Representations (2016).
24 B. M. Williams and L. A. Hoel: J. Transp. Eng. **129** (2003) 664. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664)
25 I. Sutskever, O. Vinyals, and Q. V. Le: Proc. 27th Int. Conf. Neural Information Processing Systems (2014) 3104–3112.
26 B. Yu, H. Yin, and Z. Zhu: SProc. 27th Int. Joint Conf. Artificial Intelligence (2018) 3634–3640. https://doi.org/10.24963/ijcai.2018/505
27 C. Zheng, X. Fan, C. Wang, and J. Qi: AAAI (2020) pp. 1234–1241.
28 T. Wang, J. Chen, J. L̈u, K. Liu, A. Zhu, H. Snoussi, and B. Zhang: IEEE Trans. Neural Networks and Learning Systems (2022)

About the Authors

Yuxiang Shan received his master's degree from Zhejiang University. At present, he is a technical consultant of China Tobacco Zhejiang Industry Co., Ltd. His research interests are in artificial intelligence and deep learning. (shanyuxiang123@gmail.com)

Hailiang Lu is a deputy chief of the information system operation section of China Tobacco Zhejiang Industry Co., Ltd. His research interests are in computer technology and computer applications.
(luhailiang2023@outlook.com)

Weidong Lou is a deputy director of Hangzhou Cigarette Factory, China Tobacco Zhejiang Industry Co., Ltd. His research interests are in the R&D of automation systems and computer technology. (louweidong66@outlook.com)