A Short Term Traffic Prediction Model Based on Deep Capture of Temporal Periodic Drift

Yong Liu[1], Jianxun Cui[1,2(✉)], and Zhaohua Long[3]

[1] School of Transportation Science and Engineering, Harbin Institute of Technology, Harbin, China
cuijianxun@hit.edu.cn
[2] Chongqing Research Institute of HIT, 618 Liangjiang Avenue, Longxing Town, Yubei District, Chongqing, China
[3] Chongqing Hualian Zhongzhi Technology Co., Ltd., No.21-1-205, Zhuoyue Road, Longxing Town, Yubei District, Chongqing, China

Abstract. Accurate prediction of short-term traffic flow is crucial for the control and guidance of urban traffic. This paper proposes a new deep learning traffic flow prediction model, Spatial-Temporal fusion model based on Deformable Convolution (DConv-ST), which deeply captures the spatiotemporal correlations present in the sequences. The model divides the raw data into three types of sequences containing information about recent, daily, and weekly periods. A deformable convolutional module is constructed to solve the problem of temporal periodic drift in the sequence, and graph attention network and multi-head attention mechanism are used to capture local and global spatial correlations. A gated mechanism is used to fuse the results of each component module for output. Experiments including model performance analysis, important component analysis, and ablation analysis were conducted on the publicly available transportation network datasets PeMS04 and PeMS08. The experimental results all demonstrate the superior performance of the proposed model.

Keywords: Short-term traffic flow prediction · Temporal periodic drift · Attention mechanism

1 Introduction

Short-term traffic flow prediction involves forecasting upcoming traffic conditions within a brief time period. This prediction is based on historical traffic data and relevant external factors like weather and events. It covers the short-term prediction of traffic state indicators including speed, flow, density, and congestion status. This field of research tackles fundamental and essential scientific challenges that underlie intelligent transportation applications like regional traffic coordination, route guidance, travel information dissemination, formulation of emergency traffic management strategies, and more.

X. Chen et al. (Eds.): IoTaaS 2023, LNICST 585, pp. 183–199, 2025.
https://doi.org/10.1007/978-3-031-70507-6_15

Despite the impressive predictive performance achieved by deep learning-based short-term traffic flow prediction, only a few studies have taken into consideration the fluctuations inherent in the periodic variations of historical traffic flows, referred to as the temporal periodic drift phenomenon. As an illustration, with the arrival of winter, the timing of evening rush hours gradually shifts earlier, and the peak hours on different days may not align consistently, exhibiting variations across 1-2 time intervals. Failing to comprehensively address this temporal periodic drift phenomenon hinders the potential for further enhancing the predictive performance of deep learning in short-term traffic flow prediction. Conversely, the majority of current research emphasizes capturing spatial correlations within traffic flows predominantly through the adjacency matrix defined by the road network topology. This approach involves the fusion of spatial node features. However, due to the nonlinear and intricate propagation characteristics of urban traffic flows, relying solely on the spatial correlations derived from the existing adjacency relationships defined by road network topology falls short in thoroughly characterizing the diverse and underlying associations among spatial nodes.

To address this, we propose a novel short term traffic prediction model: Spatial-Temporal fusion model based on Deformable Convolution (DConv-ST), driven by the following primary objectives:

- Our approach involved dividing the original data into three distinct sequences, each containing different temporal features. We developed deformable convolution module specifically designed for sequences with daily-periodic and weekly-periodic sequences that display temporal periodic drift. These modules consist of data deformation, linear interpolation, multiple convolutions, and gated outputs, effectively capturing the temporal periodic drift phenomenon inherent in historical traffic flows. This leads to a more robust temporal correlation between predicted and known historical traffic flow sequences.
- Concerning spatial fusion, our strategy aimed to facilitate feature fusion between non-adjacent nodes. Building upon the foundation of graph attention network, which capture local spatial correlations using the road network's adjacency matrix. We also introduced multi-head attention mechanism to capture global spatial correlations. We then integrated a gated mechanism to produce the fused output from both local and global approaches.
- A comprehensive set of experiments were conducted on actual highway traffic datasets, affirming that our model surpasses existing baselines in predictive performance. We performed experiments utilizing diverse variant models and offered in-depth explanations regarding their performance and underlying design principles.

2 Related Works

Constrained by three key factors: the depth of problem understanding, the scale of sample data, and the modeling capacity of forecasting methodologies, the initial phase of short-term traffic state prediction commenced with time series

models [9,16]. These encompassed autoregressive models, moving average models, Autoregressive Moving Average models (ARMA), historical average models, and others. As comprehension grew, the recognition of non-stationarity and nonlinearity in traffic flow sequences led to the adoption of techniques like Autoregressive Integrated Moving Average (ARIMA [20]), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), and Kalman Filtering [15] for prediction.

During the 1980s and 1990s, machine learning theory witnessed significant advancement and widespread commercial applications [10]. Pertinent theories like Support Vector Machines (SVM [4]) and Support Vector Regression (SVR [17]) were also applied to road traffic prediction. Additionally, Bayesian methods, K-Nearest Neighbors (KNN [13]), shallow neural networks, and other strategies found application in road traffic prediction.

In the contemporary context, thanks to the availability of abundant traffic big data and the maturity of deep learning theory, the utilization of deep neural networks as the foundational methodology for short-term traffic state prediction has emerged as a prominent and mainstream research pursuit. Models have widely integrated auto-encoders, diverse Convolutional Neural Networks (CNN) [6,23,25] (such as 1D causal convolutions, 2D image convolutions, 3D image convolutions), Recurrent Neural Networks (RNN) [3] (like LSTM [8,14,27], GRU [1]), hybrid convolutional and recurrent networks [24] (ConvLSTM, PredRNN), and Graph Neural Networks (GNN, including GCN [11], Diffusion Convolution [12], GAT [19]). When combined with deformable convolution networks [2], these models possess the ability to dynamically and flexibly modify the shape of convolutional kernels to accommodate deformation characteristics. Additionally, concepts from the field of neural networks, such as spatiotemporal attention mechanisms [5] and transformers [21,22], have found extensive application in the study of short-term traffic state prediction.

Current traffic flow models based on spatiotemporal fusion extract temporal and spatial features separately using graph neural networks and temporal models. Yu et al. [25] introduced a method named Spatiotemporal Graph Convolution Model (STGCN), utilizing Temporal-Spatial-Temporal (TST) triple convolution units to simultaneously capture the spatiotemporal features of road networks. The model exhibits excellent performance. Guo et al. [7] proposed ST-3DNet, incorporating 3D convolutions to capture temporal and spatial similarities in traffic data, considering short-term and long-term temporal attributes. Attention mechanisms are also integrated in several models. Zhao et al. [26] introduced STCGAT, primarily composed of node adaptive learning, graph convolutions, and local and global causal temporal convolution modules to collectively learn local and global spatiotemporal dependencies. Guo et al. [5] proposed ASTGCN, which employs three identical components to extract spatiotemporal correlations from input data spanning three distinct historical periods. The final outcomes are obtained through weighted fusion. These components consist of spatiotemporal attention mechanisms, one-dimensional convolutions, and graph convolutions.

3 Methodology

3.1 Problem Definition

Sensors within the road network can be depicted as a graph denoted by $G = (V, E, A)$, in which $V = \{v_1, v_2, \cdots, v_N\}$ signifies the node set and $|V| = N$ denotes the count of nodes; E signifies the set of edges connecting nodes; $A \in \mathbb{R}^{N \times N}$ stands as the adjacency matrix for the graph G. The traffic flow features observed on the graph G at time t are indicated as $\boldsymbol{x}^t \in \mathbb{R}^{N \times C}$, with C being the number of features per node. Assuming historical data from T time periods is utilized to predict data for the future T_p time periods, the objective is to learn a function $f(\cdot)$ to achieve

$$\left[\boldsymbol{x}^{t-T+1}, \boldsymbol{x}^{t-T}, \ldots, \boldsymbol{x}^{t}; G\right] \xrightarrow{f(\cdot)} \left[\boldsymbol{x}^{t+1}, \boldsymbol{x}^{t+2}, \ldots, \boldsymbol{x}^{t+T_p}\right] \tag{1}$$

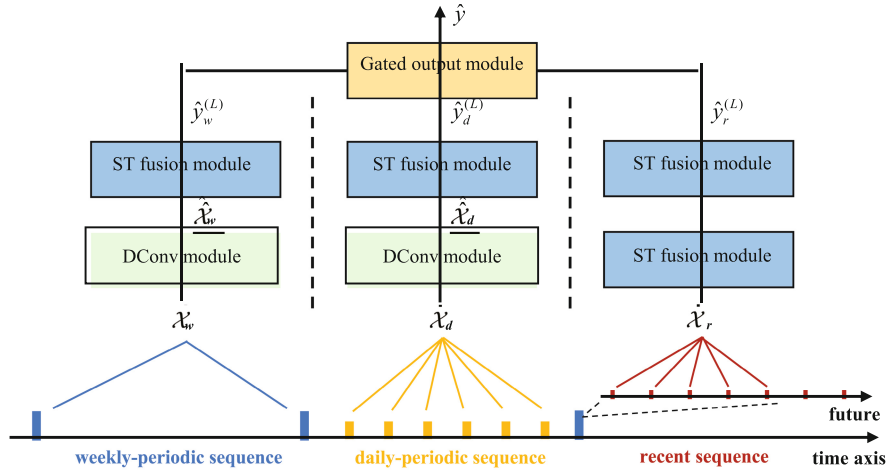
3.2 Model Architecture and Data Organization

Fig. 1 presents the overall framework of the DConv-ST model proposed in this paper. The model is comprised of three primary modules. The raw data must be structured into three distinct sequence types: the recent sequence, the daily-periodic sequence, and the weekly-periodic sequence. The model will then individually handle the modeling of these three sequence types.

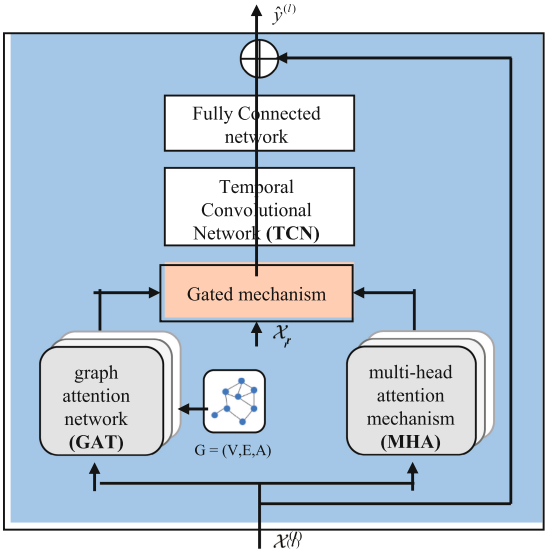
Given a sampling frequency of q times per day, the current time interval is denoted as t_0, and the historical time periods for the three sequences are labeled as T_r, T_d, and T_w, respectively. The latter two satisfy the conditions $T_d = N_d \cdot (T_p + 2T_{drift})$ and $T_w = N_w \cdot (T_p + 2T_{drift})$, where N_d and N_w correspond to the number of historical days for the daily-periodic sequence and the number of historical weeks for the weekly-periodic sequence, respectively. T_{drift} signifies the length of indirectly related time period data introduced to address the temporal periodic drift issue. The specific formats of these three time sequences are as delineated below:

(1)The recent sequence: $\boldsymbol{\mathcal{X}}_r = \left(\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^N\right) \in \mathbb{R}^{N \times T_r}$, where $\mathbf{X}^n = \left(x^n_{t_0-T_r+1}, x^n_{t_0-T_r+2}, \ldots, x^n_{t_0}\right) \in \mathbb{R}^{T_r}$.

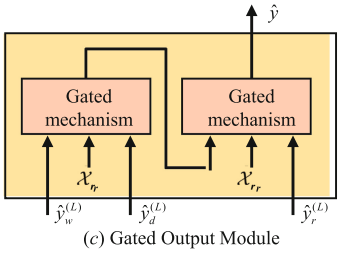
(2)The daily-periodic sequence: $\boldsymbol{\mathcal{X}}_d = \left(\mathbf{X}^1_d, \mathbf{X}^2_d, \ldots, \mathbf{X}^N_d\right)$
$\in \mathbb{R}^{N \times N_d \times (T_p + 2T_{drift})}$, where $\mathbf{X}^n_d = \left(\mathrm{X}^n_{N_d}, \mathrm{X}^n_{N_d-1}, \ldots, \mathrm{X}^n_1\right) =$
$((x^n_{t_0-N_d \cdot q-T_{drift}+1}, \ldots, x^n_{t_0-N_d \cdot q}, x^n_{t_0-N_d \cdot q+1}, \ldots, x^n_{t_0-N_d \cdot q+T_p}, x^n_{t_0-N_d \cdot q+T_p+1},$
$\ldots, x^n_{t_0-N_d \cdot q+T_p+T_{drift}})$, $(x^n_{t_0-(N_d-1) \cdot q-T_{drift}+1}, \ldots, x^n_{t_0-(N_d-1) \cdot q}, x^n_{t_0-(N_d-1) \cdot q+1},$
$\ldots, x^n_{t_0-(N_d-1) \cdot q+T_p}, x_{t_0-(N_d-1)}, \ldots,$
$x^n_{t_0-(N_d-1) \cdot q+T_p+T_{drift}}), \ldots, (x^n_{t_0-q-T_{drift}+1},$
$\ldots, x^n_{t_0-q}, x^n_{t_0-q+1}, \ldots, x^n_{t_0-q+T_p}, x^n_{t_0-q+T_p+1}, \ldots, x^n_{t_0-q+T_p+T_{drift}}))$
$\in \mathbb{R}^{N_d \times (T_p + 2T_{drift})}$. $\mathrm{X}^n_{N_d}$ denotes the data of node n from N_d days ago. The subscripts span from $(t_0 - N_d \cdot q - T_{drift} + 1)$ to $(t_0 - N_d \cdot q)$ and from $(t_0 - N_d \cdot q + T_p + 1)$ to $(t_0 - N_d \cdot q + T_p + T_{drift})$, representing indirectly related time period data at both ends with a length of T_{drift}. The subscripts from $(t_0 - N_d \cdot q + 1)$ to $(t_0 - N_d \cdot q + T_p)$ correspond to directly related time period



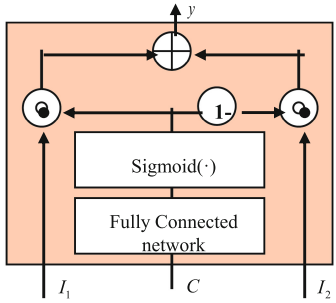
(*a*) Framework of DConv-ST



(*b*) Spatiotemporal Fusion Module



(*c*) Gated Output Module



(*d*) Gated Mechanism

Fig. 1. The overall framework and main modules of DConv-ST. (a) The framework of DConv-ST. The three main modules are: the spatiotemporal fusion module (ST fusion module), the deformable convolution module (DConv module), and the Gated output module. (b) Spatiotemporal fusion module, including spatial fusion and temporal fusion. (c) Gated output module, which fuses the results of three sequences. (d) Gated Mechanism, used to implement gated output module.

data with a length of T_p. The representation of data for other historical days follows a similar pattern.

(3)The weekly-periodic sequence: $\boldsymbol{\mathcal{X}}_w = \left(\mathbf{X}_w^1, \mathbf{X}_w^2, \ldots, \mathbf{X}_w^N\right) \in \mathbb{R}^{N \times N_w \times (T_p + 2T_{drift})}$, where $\mathbf{X}_w^n = \left(\mathrm{X}_{N_w}^n, \mathrm{X}_{N_w-1}^n, \ldots, \mathrm{X}_1^n\right)$. Its form is similar to the daily-periodic sequence.

It's crucial to emphasize that the structured data deviates from the depiction in Eq.1 in two distinct manners: firstly, there's a reversal in the order of dimensions between the number of time periods and the number of nodes; secondly, the dimension of feature number C is fixed at 1, signifying that the model learns and predicts only one feature (flow, speed, density) at a time, necessitating the establishment of distinct models for different features. Fig. 2 visually illustrates the process of organizing raw data into a set of data.

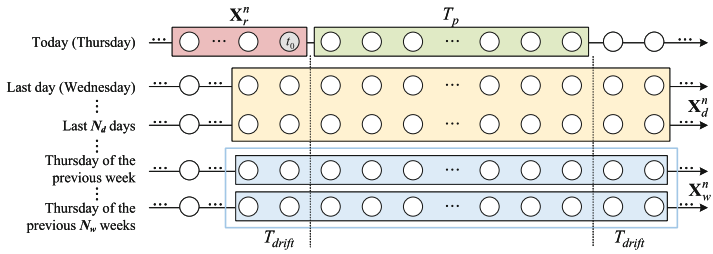


Fig. 2. In each data set, the subsequent T_p data points of the current time period are taken as the prediction target. The input data consists of three different types of historical time period data: **Recent Sequence**, which involves the time period data from the last T_r data points leading up to the current time period. **Daily-Periodic Sequence**, which involves the time period data from the same time period on the days ranging from 1 day ago to N_d days ago. **Weekly-Periodic Sequence**, which involves the time period data from the same day and time period in the weeks ranging from 1 week ago to N_w weeks ago. Furthermore, on both sides of the daily-periodic and weekly-periodic sequence, an additional segment of length T_{drift} is appended, which is utilized to address temporal periodic drift issues.

3.3 Deformable Convolution Module

The role of the deformable convolution module serves a dual purpose. Firstly, it tackles the challenge of temporal periodic drift within the sequences by modifying the dimensions and structures of the daily-periodic and the weekly-periodic sequence, ensuring their smooth incorporation through the spatiotemporal fusion module. Secondly, this module can capture the spatiotemporal traits intrinsic to these two sequences, setting them apart from the recent sequence. For instance, it has the ability to capture the periodic patterns existing between distinct days and weeks.

Taking the deformable convolution process of the daily-periodic sequence as an example (as shown in Fig. 3). First, the input sequence undergoes a deformation process. Let's use the data of one node as an illustration:

$$\mathbf{X}_{d,p}^{n(deform)} = \sum_{p_n \in \mathcal{R}} h\left(p_n, T_{drift} \cdot \sigma_t\left(M_{d,p}\right)\right) \cdot \mathbf{X}_{d,p+T_{drift}+p_n}^{n} \tag{2}$$

where $\mathcal{R} = \{-T_{drift}, -(T_{drift}-1), \ldots, -1, 0, 1, \ldots, T_{drift}-1, T_{drift}\}$, $d = 1, 2, \ldots, N_d$, and $p = 1, 2, \ldots, T_p$. The matrix $M \in \mathbb{R}^{N_d \times T_p}$ serves as the offset matrix. Initially, all its elements are initialized to 0, and they are subsequently updated during model learning, constituting the parameters that the model must acquire. Denoted as $\sigma_t(\cdot)$, the activation function is chosen as the hyperbolic tangent function $\tanh(\cdot)$, generating outputs within the range of -1 to 1. $h(\cdot,\cdot)$ stands for the linear interpolation function, defined as $h(a,b) = \max(0, 1-|a-b|)$.

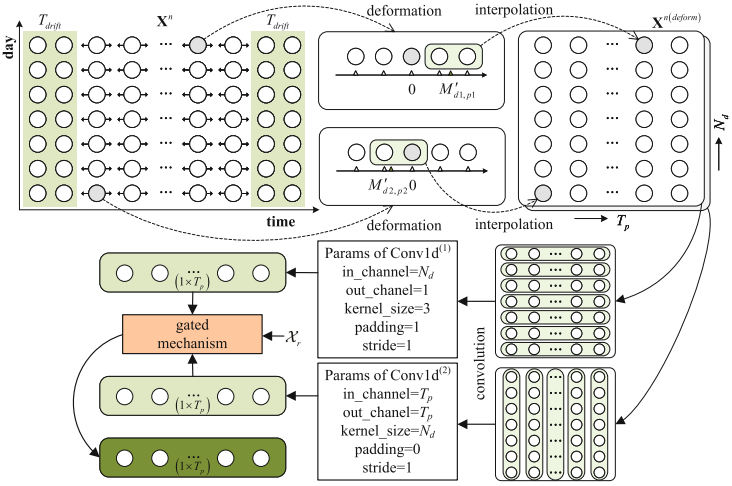


Fig. 3. An example diagram of the deformable convolution process for the daily-periodic sequence. It mainly includes four operations: deformation, interpolation, convolution, and gated output.

The process mentioned above is depicted in the upper section of Fig. 3, where $M'_{d,p} = T_{drift} \cdot \sigma_t(M_{d,p})$, and each $M'_{d,p}$ is confined within the range of $(-T_{drift}, T_{drift})$. By interpolating the data corresponding to the positions resulting from rounding this value both upwards and downwards, the modified result is obtained. Through the process of deformation and interpolation, the dimensions of the input data transform from $\mathbf{X}^n \in \mathbb{R}^{N_d \times (T_p + 2T_{drift})}$ to $\mathbf{X}^{n(deform)} \in \mathbb{R}^{N_d \times T_p}$, incorporating the relevant segment of indirectly related time period data into the directly related time period data. This procedure contributes to mitigating the temporal periodic drift issue inherently present in the sequence, to some extent.

The deformed sequence yields a two-dimensional array denoted as $\mathbf{X}^{n(deform)}$. Along its dimension of length T_p, it represents traffic feature data from historical days that correspond to the same time intervals as the prediction period, after addressing the temporal periodic drift issue. Hence, one-dimensional convolution is employed to capture the changing characteristics of the time sequence. Along its dimension of length N_d, it represents traffic data for the same time intervals across different days. Recognizing that traffic flow characteristics between distinct days can exhibit fluctuations and divergent relevance levels to the features of the target period, one-dimensional convolution is employed to attribute distinct weights to different days. The process of implementation is outlined as follows:

$$\boldsymbol{\mathcal{X}}_d^{(d)} = \|_{n=1}^{N} \mathrm{Conv1d}^{(1)} \left(\|_{d=1}^{N_d} \mathbf{X}_{d,:}^{n(deform)} \right) \tag{3}$$

$$\boldsymbol{\mathcal{X}}_d^{(p)} = \|_{n=1}^{N} \mathrm{Conv1d}^{(2)} \left(\|_{p=1}^{T_p} \mathbf{X}_{:,p}^{n(deform)} \right) \tag{4}$$

where $\|$ signifies the concatenation of outputs. The chosen parameters for one-dimensional convolution and linear weighting must ensure that the dimensions of the results satisfy $\boldsymbol{\mathcal{X}}_d^{(d)}, \boldsymbol{\mathcal{X}}_d^{(p)} \in \mathbb{R}^{N \times T_p}$. Ultimately, the integration of the two results occurs through a gated mechanism, outlined as follows:

$$G\left(C, I_1, I_2\right) = I_1 \odot \sigma_s(C \cdot W_g + b_g) + I_2 \odot (1 - \sigma_s(C \cdot W_g + b_g)) \tag{5}$$

where I_1 and I_2 denote the inputs, representing the two sets of data to be fused, while C is the control parameter. The symbol $\odot$ indicates the Hadamard product. The activation function $\sigma_s(\cdot)$ is chosen to be the sigmoid function. The matrices $W_g \in \mathbb{R}^{T_r \times T_p}$ and $b_g \in \mathbb{R}^{N \times T_p}$ are utilized for linearly weighting the control parameter. These matrices are dynamically adjusted through model learning to regulate the two input sets. Utilizing $\boldsymbol{\mathcal{X}}_r$ for control, the final output after gating is as follows:

$$\hat{\boldsymbol{\mathcal{X}}}_d = G\left(\boldsymbol{\mathcal{X}}_r, \boldsymbol{\mathcal{X}}_d^{(d)}, \boldsymbol{\mathcal{X}}_d^{(p)}\right) \tag{6}$$

3.4 Spatiotemporal Fusion Module

The spatiotemporal fusion module consists of two main components: spatial fusion and temporal fusion. Let the input for the l-th spatiotemporal fusion module be denoted as $\boldsymbol{\mathcal{X}}^{(l)} \in \mathbb{R}^{N \times T_{in}^{(l)}}$, and its output as $\hat{\boldsymbol{y}}^{(l)} \in \mathbb{R}^{N \times T_{out}^{(l)}}$. This implies that $\boldsymbol{\mathcal{X}}^{(l+1)} = \hat{\boldsymbol{y}}^{(l)}$ and $T_{in}^{(l+1)} = T_{out}^{(l)}$. The three sets of sequences passing through the spatiotemporal fusion modules share the same structure. Specifically, the inputs entering the first spatiotemporal fusion module are $\boldsymbol{\mathcal{X}}_r^{(1)} = \boldsymbol{\mathcal{X}}_r \in \mathbb{R}^{N \times T_r}$, $\boldsymbol{\mathcal{X}}_d^{(1)} = \hat{\boldsymbol{\mathcal{X}}}_d \in \mathbb{R}^{N \times T_p}$, and $\boldsymbol{\mathcal{X}}_w^{(1)} = \hat{\boldsymbol{\mathcal{X}}}_w \in \mathbb{R}^{N \times T_p}$.

In the l-th spatiotemporal fusion module, let the inputs and outputs of the spatial and temporal fusion modules be represented as $\boldsymbol{\mathcal{X}}^{s(l)}$, $\hat{\boldsymbol{y}}^{s(l)}$, $\boldsymbol{\mathcal{X}}^{t(l)}$, and $\hat{\boldsymbol{y}}^{t(l)}$ respectively. This leads to $\boldsymbol{\mathcal{X}}^{s(l)} = \boldsymbol{\mathcal{X}}^{(l)}$ and $\hat{\boldsymbol{y}}^{t(l)} = \hat{\boldsymbol{y}}^{(l)}$. Since the spatial and temporal fusion modules are directly linked, we have $\boldsymbol{\mathcal{X}}^{t(l)} = \hat{\boldsymbol{y}}^{s(l)}$, with its dimension set as $\mathbb{R}^{N \times T_{hid}^{(l)}}$.

Spatial Fusion Module Within the spatial fusion module, both graph attention network(GAT) and multi-head attention mechanism(MHA) are utilized to aggregate distinctively the local and the global spatial correlations of the graph.

Within the framework of GAT , node features are merged based on the adjacency matrix associated with each node, enabling the capture of immediate spatial correlations among nodes. The procedural execution is outlined as follows: For the node i in the graph, calculate the similarity coefficients $S_{i,j}$ and the attention coefficient matrix S' between it and its neighboring nodes $\mathcal{N}_i$ one by one:

$$S_{i,j} = [\boldsymbol{h}_i \| \boldsymbol{h}_j] \cdot a, j \in \mathcal{N}_i \tag{7}$$

$$S'_{i,j} = \begin{cases} \frac{\exp(\text{LeakyReLU}(S_{i,j}))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(S_{i,k}))} & j \in \mathcal{N}_i \\ 0 & j \notin \mathcal{N}_i \end{cases} \tag{8}$$

where $\boldsymbol{h}$ denotes the input sequence following feature enhancement and $\boldsymbol{h} = \boldsymbol{\mathcal{X}}^{s(l)} W_a \in \mathbb{R}^{N \times T'}$. The operation $[\cdot \| \cdot]$ concatenates the features transformed from node i and node j. Subsequently, through the mapping a, a scalar is generated, representing the similarity coefficient between them. This coefficient is then employed to calculate the attention coefficient matrix $S' \in \mathbb{R}^{N \times N}$. Then, the outcome of feature fusion is achieved by weighting $\boldsymbol{h}$ using this matrix:

$$g\left(\boldsymbol{\mathcal{X}}^{s(l)}, \mathcal{N}\right) = \|_{i=1}^{N} \left(S'_{i,:} \cdot \boldsymbol{h} + b_a\right) \in \mathbb{R}^{N \times T'} \tag{9}$$

where $b_a \in \mathbb{R}^{1 \times T'}$. With the introduction of the Multi-Head concept, a spatial fusion module incorporates K graph attention networks, and the resulting outputs are as depicted:

$$\hat{\boldsymbol{y}}_{gat}^{(l)} = g\left(\|_{k=1}^{K} g_{(k)}\left(\boldsymbol{\mathcal{X}}^{s(l)}, \mathcal{N}\right), \mathcal{N}\right) \in \mathbb{R}^{N \times T_{hid}^{(l)}} \tag{10}$$

where $g_{(k)}(\cdot, \cdot)$ corresponds to the network at the k-th head. Each distinct head within the network undergoes training using separate parameter sets W_a and a. The concatenated outcomes from all these individual heads result in a dimensionality of $\mathbb{R}^{N \times KT'}$. This combined output is employed as input for an additional GAT, where T' is configured to match $T_{hid}^{(l)}$ within this specific network.

MHA dispenses with the necessity for a predefined adjacency matrix; it can dynamically learn a matrix for the purpose of fusion. It conducts fusion based on the pairwise similarity between each node and others, thereby capturing global and latent spatial correlations. The procedural sequence is outlined as follows: Given the input $\boldsymbol{\mathcal{X}}^{s(l)}$, three distinct subspaces are derived: the query subspace $Q \in \mathbb{R}^{N \times t_q}$, the key subspace $K \in \mathbb{R}^{N \times t_k}$, and the value subspace $V \in \mathbb{R}^{N \times t_v}$. The method to obtain them is elucidated as follows:

$$Q = \boldsymbol{\mathcal{X}}^{s(l)} W^Q, K = \boldsymbol{\mathcal{X}}^{s(l)} W^K, V = \boldsymbol{\mathcal{X}}^{s(l)} W^V \tag{11}$$

The output obtained through the calculation using Scaled Dot-Product Attention is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{t_k}}\right) V \tag{12}$$

It's important to highlight that the dimensions of Q and K must fulfill the condition $t_q = t_k$. When incorporating the notion of multi-head, the resulting output from MHA is as follows:

$$\hat{\boldsymbol{y}}_{mha}^{(l)} = \left(\|_{h=1}^{H} \text{head}_i\right) W^O \in \mathbb{R}^{N \times T_{hid}^{(l)}} \tag{13}$$

where each head, denoted as $\text{head}_i = \text{Attention}\left(Q_i, K_i, V_i\right)$, operates independently. Also, the matrix $W^O \in \mathbb{R}^{H \cdot t_v \times T_{hid}^{(l)}}$ is selected to ensure the output dimension matches that of GAT.

Finally, resembling the gated mechanism used in the deformable convolution module, the linearly weighted outcome $\boldsymbol{\mathcal{X}}_r$ is utilized to regulate the fusion process between the outputs of GAT and MHA. This procedure is executed as outlined below:

$$\hat{\boldsymbol{y}}^{s(l)} = G_s\left(\boldsymbol{\mathcal{X}}_r, \hat{\boldsymbol{y}}_{gat}^{(l)}, \hat{\boldsymbol{y}}_{mha}^{(l)}\right) \tag{14}$$

where the implementation of the gated mechanism $G_s\left(\cdot,\cdot,\cdot\right)$ is presented in Eq.5, with distinct parameters W_g and b_g that are not shared with other gated mechanisms.

Temporal Fusion Module The temporal fusion module predominantly employs a Temporal Convolutional Network (TCN) for its execution. For a specific node data $\boldsymbol{\mathcal{X}}_{n,:}^{t(l)} \in \mathbb{R}^{T_{hid}^{(l)}}$ within the input sequence of this module, and a set of filters $F = \{f_1, f_2, \ldots, f_K\}$, the dilated convolution with dilation factor d at position $\boldsymbol{\mathcal{X}}_{n,\tau}^{t(l)}$ within the sequence $\boldsymbol{\mathcal{X}}_{n,:}^{t(l)}$ is expressed as:

$$\left(F *_d \boldsymbol{\mathcal{X}}_n^{t(l)}\right)(\tau) = \sum_{k=1}^{K} f_k \cdot \boldsymbol{\mathcal{X}}_{n,\tau-d(K-k)}^{t(l)} \tag{15}$$

Each node's sequence is individually processed by a TCN, and then passed through two linear layers to more comprehensively incorporate historical temporal information.

3.5 Gated Output Module

After each of the three temporal sequences has passed through the spatiotemporal fusion module for all L layers, the outputs $\hat{\boldsymbol{y}}_r^{(L)}$, $\hat{\boldsymbol{y}}_d^{(L)}$, and $\hat{\boldsymbol{y}}_w^{(L)}$ undergo fusion via the gated mechanism. Since gating typically involves two data sets, the gating fusion process is initially applied to $\hat{\boldsymbol{y}}_d^{(L)}$ and $\hat{\boldsymbol{y}}_w^{(L)}$, owing to their shared characteristics - both having been subject to processing through the deformable convolution module and possessing historical cyclic features. Subsequently, the outcome of this fusion is further combined through gating with $\hat{\boldsymbol{y}}_r^{(L)}$. Similarly, the adjacent temporal sequences $\boldsymbol{\mathcal{X}}_r$ are employed as control values for these two gating operations. The ultimate output of the entire model, denoted as $\hat{\boldsymbol{y}}$, is then derived as follows:

$$\hat{\boldsymbol{y}} = G_{o2}\left(\boldsymbol{\mathcal{X}}_r, \hat{\boldsymbol{y}}_r^{(L)}, G_{o1}\left(\boldsymbol{\mathcal{X}}_r, \hat{\boldsymbol{y}}_d^{(L)}, \hat{\boldsymbol{y}}_w^{(L)}\right)\right) \tag{16}$$

where the definitions of the gating functions $G_{o1}(\cdot,\cdot,\cdot)$ and $G_{o2}(\cdot,\cdot,\cdot)$ can be found in Eq.5. The linear weighting parameters, W_{o1} and W_{o2}, belong to the space $\mathbb{R}^{T_{out}^{(L)} \times T_p}$, while b_{o1} and b_{o2} belong to $\mathbb{R}^{N \times T_p}$.

4 Experiment

In order to assess the effectiveness of DConv-ST, a range of experiments were formulated. These encompassed evaluating the performance of the DConv-ST model itself, conducting experiments with variant models, and performing ablation analyses. These experiments were conducted on two highway traffic datasets, PeMS04 and PeMS08, both originating from California, USA.

4.1 Datasets

The data for PeMS04 is derived from 3,848 sensors situated along 29 roadways within the San Francisco Bay Area. This dataset covers the time span from January 1st to February 28th, 2018. The data for PeMS08, on the other hand, is collected from 1,979 sensors positioned along 8 roadways in San Bernardino, spanning from July 1st to August 31st, 2016. Following the elimination of redundant sensors, each dataset retains information from 307 and 170 sensors respectively. Additionally, these datasets incorporate adjacency details among the sensors. The original data is aggregated into 5-minute intervals, capturing three key attributes: total flow, average speed, and average occupancy.

4.2 Settings

When it comes to data processing, linear interpolation is employed to address missing data. The selected predictive feature is traffic flow. When structuring the data into the desired sequence format, the parameters are configured as follows: $T_r = 12$, $N_d = 7$, $N_w = 4$. The introduction of indirectly related time period data brings a length of $T_{drift} = 3$, while the prediction interval extends for $T_p = 12$, signifying the aim to forecast traffic flow for the upcoming hour. Following this organization, the PeMS04 dataset yields a total of 8914 data sets, and the PeMS08 dataset produces 9778 data sets. The dataset is partitioned into three segments, allocating 60% for training, 20% for validation, and 20% for testing. Standardization is implemented on the sequences $\boldsymbol{\mathcal{X}}_r$, $\boldsymbol{\mathcal{X}}_d$, and $\boldsymbol{\mathcal{X}}_w$ within each subset using the formula $x' = (x - \text{mean}(x))/\text{std}(x)$.

The DConv-ST model is instantiated within the PyTorch framework. For this model, the layers in the spatiotemporal fusion module are configured as 2 for $\boldsymbol{\mathcal{X}}_r$, 1 for $\boldsymbol{\mathcal{X}}_d$, and 1 for $\boldsymbol{\mathcal{X}}_w$ sub-models, respectively. In the last two sub-models, the deformable convolution module consists of a single layer. Throughout the model training process, the Mean Squared Error (MSE) serves as the loss function, and the backpropagation algorithm is applied to minimize the discrepancy between predicted and true values. During training, the batch size is 16, and the learning rate is 0.001.

4.3 Baselines

We compare DConv-ST with the following ten baselines:

- HA: Historical Average method.
- ARIMA : Autoregressive Integrated Moving Average method.
- VAR: Vector Autoregression model. It can capture relationships among all traffic flow sequences.
- LSTM [8]: Long Short-Term Memory network. It's a special type of gated RNN model.
- GRU [1]: Gated Recurrent Unit network. It incorporates learnable gate mechanisms and is a variation of LSTM.
- DCRNN [12]: Diffusion Convolutional Recurrent Neural Network. It simulates the information diffusion process using random walks on graphs and employs gated recurrent units to build a sequence-to-sequence model.
- STGCN [25]: Spatio-Temporal Graph Convolutional Network. It replaces RNN with 1D causal convolutions for temporal dependencies and employs spectral graph convolutions for spatial dependencies.
- STSGCN [18]: Spatio-Temporal Synchronous Graph Convolutional Network. It introduces a synchronous modeling mechanism to capture complex local spatio-temporal dependencies.
- MSTGCN, ASTGCN [5]: The latter is Attention-based Spatio-Temporal Graph Convolutional Network. MSTGCN removes the spatio-temporal attention module from ASTGCN.

Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are chosen as evaluation metrics in the experiments.

4.4 Experimental Results

Performance Comparison The performance of DConv-ST was assessed against ten baselines using the PeMS04 and PeMS08 datasets. Table 1 showcases the RMSE and MAE outcomes for each model when forecasting traffic flow for the upcoming hour (12 time intervals, excluding HA). It is evident that the proposed DConv-ST model surpasses all other models across various evaluation metrics on both datasets. In comparison to traditional time series analysis methods, deep learning-based models exhibit superior overall performance and stronger feature capturing capabilities, making them more advantageous in handling highly nonlinear and complex traffic data. Particularly noteworthy are models that account for both spatial and temporal correlations, encompassing the last four baselines and the DConv-ST model introduced in this study. They consistently outperform traditional deep learning models such as LSTM and GRU. In contrast to ASTGCN, which also employs attention mechanisms, our model showcases reduced errors. This is partially attributed to the incorporation of deformable convolution modules, augmenting the model's precision in capturing spatiotemporal features.

Table 1. The performance of the 10 selected baselines and DConv-ST on the PeMS04 and PeMS08 datasets. DConv-ST model surpasses all other baselines across various evaluation metrics on both datasets.

Models	PeMS04		PeMS08	
	RMSE	MAE	RMSE	MAE
HA	54.14	36.76	44.03	29.52
ARIMA	68.13	32.11	43.30	24.04
VAR	51.73	33.76	31.21	21.41
LSTM	45.82	29.45	36.96	23.18
GRU	45.11	28.65	35.95	22.20
DCRNN	38.12	24.70	28.83	19.86
STGCN	35.55	22.70	27.87	18.88
STSGCN	33.65	21.19	26.80	17.13
MSTGCN	35.64	22.73	26.47	17.47
ASTGCN	32.82	21.80	25.27	16.63
DConv-ST (ours)	**31.59**	**19.78**	**24.49**	**15.71**

Component Analysis The performance of deformable convolution modules and spatial fusion methods was investigated on the PeMS04 dataset.

Influence of Deformable Convolution Module In order to investigate the influence of deformable convolution modules on model performance, three sets of control experiments were carried out. The outcomes are depicted in Table 2. The incorporation of deformable convolution module leads to a reduction in prediction errors. Specifically, for the $\boldsymbol{\mathcal{X}}_d$ sub-model, the RMSE and MAE decreased by 4.64% and 4.47% respectively. In the case of the $\boldsymbol{\mathcal{X}}_w$ sub-model, these two metrics experienced minimal decreases, amounting to 0.89% and 1.42% respectively. The substitution of deformable convolution modules within the DConv-ST model corroborates this finding, resulting in respective metric reductions of 1.77% and 2.75%. When comparing these outcomes to the predictions of ASTGCN in Table 1, the errors of the fusion model utilizing standard convolutions closely align. This observation partially suggests that relying solely on attention mechanisms and convolution operations isn't sufficient for achieving further reductions in model errors. The deformable convolution module we devised contribute to an improved precision in traffic flow prediction results.

Methods of Spatial Fusion An investigation was conducted into the spatial fusion methods using GAT and MHA, and the results are presented in Table 3. Observing the results from the sub-models, it becomes apparent that the global spatial fusion using MHA tends to yield superior performance in the majority of cases. In contrast to the local spatial fusion accomplished by GAT, the global fusion process entails the consideration of features from all nodes within the road network. However, when dealing with substantial amounts of data, this approach

Table 2. Experimental results on the impact of deformable convolution modules on model performance. The first four rows present outcomes derived solely from the $\boldsymbol{\mathcal{X}}_d$ and $\boldsymbol{\mathcal{X}}_w$ sub-models, whereas the last two rows exhibit results generated by the fusion model. The terms (conv) and (dconv) signify whether the model utilizes conventional convolution modules (one-dimensional convolutions) or deformable convolution modules.

Models	RMSE	MAE
$\boldsymbol{\mathcal{X}}_d$(conv)	34.90	22.35
$\boldsymbol{\mathcal{X}}_d$(dconv)	33.28	21.35
$\boldsymbol{\mathcal{X}}_w$(conv)	33.80	21.74
$\boldsymbol{\mathcal{X}}_w$(dconv)	33.50	21.43
$\boldsymbol{\mathcal{X}}_r$ - $\boldsymbol{\mathcal{X}}_d$ (conv) - $\boldsymbol{\mathcal{X}}_w$(conv)	32.16	20.34
$\boldsymbol{\mathcal{X}}_r$ - $\boldsymbol{\mathcal{X}}_d$ **(dconv) - $\boldsymbol{\mathcal{X}}_w$(dconv)** (**DConv-ST**)	**31.59**	**19.78**

might encounter difficulties in accurately capturing crucial information, potentially leading to comparatively diminished performance within the fusion model. Analyzing the experimental results of the fusion model reveals that simultaneous utilization of both spatial fusion methods leads to a reduction of 2.11% and 8.06% in RMSE compared to models that exclusively employ GAT or MHA for spatial fusion, respectively.

Ablation Study Additionally, the effectiveness of individual components within the DConv-ST model was verified through ablation experiments, and the resulting RMSE and MAE outcomes are visualized in Fig. 4. Models that exclusively utilize recent sequences, lacking daily-period and weekly-period information, demonstrate the weakest performance across all models. Their performance steeply declines as the prediction interval increases, reflecting the fact that the enhanced historical information is contained within daily-period and weekly-period sequences. The outcomes of the other three ablation experiments consistently exhibit improved short-term predictive capabilities. However, as the prediction interval extends, their stability diminishes. Notably, when the prediction interval reaches 60 minutes, their performance distinctly diverges from that of DConv-ST. Our DConv-ST model integrates recent, daily-period and weekly-period sequences. It addresses temporal periodic drift issues through deformable convolution modules, and captures both global and local spatial correlations among road network nodes. This holistic approach enhances the accuracy and resilience of prediction outcomes.

Table 3. Experimental results on the impact of spatial fusion methods on model performance. The terms (gat) and (mha) respectively denote models that solely utilize GAT local spatial fusion and MHA global spatial fusion, and (gat & mha) represents the model that simultaneously employs both spatial fusion methods.

	Models	RMSE	MAE
Sub-models	$\boldsymbol{\mathcal{X}}_r$(gat)	39.96	26.11
	$\boldsymbol{\mathcal{X}}_r$(mha)	38.17	24.60
	$\boldsymbol{\mathcal{X}}_r$(**gat & mha**)	34.47	22.31
	$\boldsymbol{\mathcal{X}}_d$(gat)	34.71	22.19
	$\boldsymbol{\mathcal{X}}_d$(mha)	34.36	21.90
	$\boldsymbol{\mathcal{X}}_d$(**gat & mha**)	33.28	21.35
	$\boldsymbol{\mathcal{X}}_w$(gat)	32.22	20.22
	$\boldsymbol{\mathcal{X}}_w$(mha)	33.12	20.90
	$\boldsymbol{\mathcal{X}}_w$(**gat & mha**)	32.84	20.72
Fusion models	$\boldsymbol{\mathcal{X}}_r$ - $\boldsymbol{\mathcal{X}}_d$ - $\boldsymbol{\mathcal{X}}_w$ (gat)	32.27	20.23
	$\boldsymbol{\mathcal{X}}_r$ - $\boldsymbol{\mathcal{X}}_d$ - $\boldsymbol{\mathcal{X}}_w$ (mha)	34.36	21.90
	$\boldsymbol{\mathcal{X}}_r$ - $\boldsymbol{\mathcal{X}}_d$ - $\boldsymbol{\mathcal{X}}_w$ (**gat & mha**) (**DConv-ST**)	**31.59**	**19.78**

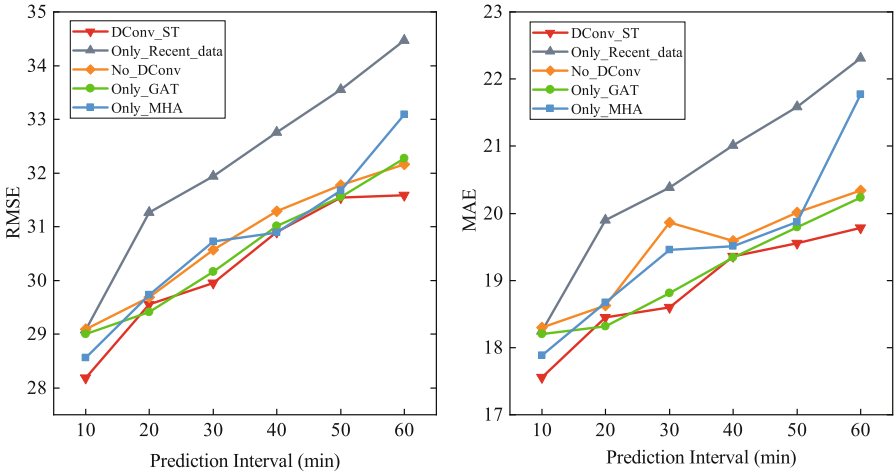


Fig. 4. The graph depicting RMSE and MAE results obtained from the ablation experiments. The experiments encompassed five scenarios: using only recent sequences (Only_Recent_data), omitting deformable convolution module (No_DConv), using only GAT for local spatial fusion (Only_GAT), using only MHA for global spatial fusion (Only_MHA), and using the complete DConv-ST model. Training and multi-step traffic flow prediction were executed using the PeMS04 dataset.

5 Conclusion

In this paper, we construct the DConv-ST short-term traffic flow prediction model. The model mainly consists of three modules: the deformable convolution module, the spatiotemporal fusion module, and the gated output module. Among them, the deformable convolution module is used to address the issue of temporal periodic drift in sequences. The spatiotemporal fusion module employs graph attention networks, multi-head attention mechanisms, and temporal convolutional networks to capture deep latent spatiotemporal correlations. The gated output module combines the results of the three types of sequences for fused output. Experimental results demonstrate that the DConv-ST model outperforms the baseline models, with the deformable convolution module and spatiotemporal fusion module playing crucial roles in enhancing model performance.

References

1. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
2. Dai, J., Qi, H. et al.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
3. Elman, J.L.: Finding structure in time. Cogn. Sci. **14**(2), 179–211 (1990)
4. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. Adv. Comput. Math. **13**, 1–50 (2000)
5. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proc. AAAI Conf. Artif. Int. **33**, 922–929 (2019)
6. Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial-temporal 3d convolutional neural networks for traffic data forecasting. IEEE Trans. Intell. Transp. Syst. **20**(10), 3913–3926 (2019)
7. Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial-temporal 3d convolutional neural networks for traffic data forecasting. IEEE Trans. Intell. Transp. Syst. **20**(10), 3913–3926 (2019)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Kamarianakis, Y., Prastacos, P.: Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches. Transp. Res. Rec. **1857**(1), 74–84 (2003)
10. Karlaftis, M.G., Vlahogianni, E.I.: Statistical methods versus neural networks in transportation research: differences, similarities and some insights. Trans. Res. Part C: Emerging Technol. **19**(3), 387–399 (2011)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
13. Luo, X., Li, D., Yang, Y., Zhang, S., et al.: Spatiotemporal traffic flow prediction with KNN and ISTM. J. Adv. Trans. **2019**(1), 4145353 (2019)
14. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Trans. Res. Part C: Emerging Technol. **54**, 187–197 (2015)

15. Mir, Z.H., Filali, F.: An adaptive kalman filter based traffic prediction algorithm for urban road network. In: 2016 12th International Conference on Innovations in Information Technology (IIT). pp. 1–6. IEEE (2016)
16. Rakha, H., Van Aerde, M.: Statistical analysis of day-to-day variations in real-time traffic flow data. Transportation research record pp. 26–34 (1995)
17. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**, 199–222 (2004)
18. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI conference on artificial intelligence. **34**, 914–921 (2020)
19. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Stat. Graph attention networks **1050**(20), 10–48550 (2017)
20. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal Arima process: theoretical basis and empirical results. J. Transp. Eng. **129**(6), 664–672 (2003)
21. Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)
22. Yan, H., Ma, X., Pu, Z.: Learning dynamic and hierarchical traffic spatiotemporal features with transformer. IEEE Trans. Intell. Transp. Syst. **23**(11), 22386–22399 (2021)
23. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
24. Yao, H. et al.: Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
25. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)
26. Zhao, W., Zhang, S., Zhou, B., Wang, B.: Stcgat: A spatio-temporal causal graph attention network for traffic flow prediction in intelligent transportation systems. arXiv preprint arXiv:2203.10749 (2022)
27. Zhao, Z., Chen, W., Wu, X., Chen, P.C., Liu, J.: LSTM network: a deep learning approach for short-term traffic forecast. IET Intel. Transport Syst. **11**(2), 68–75 (2017)