An Improved Dynamic Chebyshev Graph Convolution Network for Traffic Flow Prediction with Spatial-Temporal Attention

Lyuchao Liao, Zhiyuan Hu, Yuxin Zheng, Shuoben Bi, Fumin Zou, Huai Qiu, Maolin Zhang

Abstract

Accurate traffic flow prediction plays a significant role in urban traffic management, including traffic congestion control and public travel route planning. Recently, several approaches have been put forward to learn the patterns from historical traffic data. However, there exist some limitations resulting from the use of the static learning method to explore the dynamical characteristics of the road network. Besides, the dynamic global temporal and spatial properties are not considered in these models. These drawbacks lead to a low prediction performance and make applying to a more extensive road network challenging. To address these issues, from the inspiration of Chebyshev polynomial, we proposed an improved dynamic Chebyshev graph convolution neural network model called iDCGCN. In the proposed approach, a novel updating method for the Laplacian matrix, which approximately constructs features from different period data, is proposed based on the attention mechanism. In addition, a novel feature construction method is proposed to integrate long-short temporal and local-global spatial features for complex traffic flow representation. Experimental results have shown that iDCGCN outperforms the state-of-the-art GCN-based methods on four real-world highway traffic datasets.

Keywords: Traffic Flow Prediction, Dynamic Graph Convolution Network**,** Attention Mechanism, Spatial-temporal Convolution.

1. Introduction

Accurate and stable traffic flow forecasting plays a significant role in managing urban traffic, such as traffic congestion control and travel routes planning. Recently, the whole world is suffering from growing traffic congestion, especially in metropolitan areas [1]. The continuous aggravation of traffic congestion led to higher traffic costs, lower traffic efficiencies, and more environmental pollutions, which seriously affected people's standard of living [2]. In fact, In Intelligent Traffic System (ITS), massive traffic data could be collected by various sensors and GPS-based (Global Position Systems) equipment [3]. There is abound of rich, dynamic information hidden in these big data. However, capturing dynamic and implicit knowledge to improve accurate prediction remains a challenge in academic and industrial circles. A precise prediction could improve the travel experience and reduce the negative effects of traffic congestion, such as time consumption and economic losses [4].

Traffic flow prediction aims to forecast the specific traffic flow for travel experience and traffic management, arousing a considerable number of related studies to improve prediction accuracy. The existing challenge is mainly due to the hardness of integrating comprehensive influencing factors, including traffic accidents, weather conditions, road conditions, etc. Furthermore, it is difficult to extract the potential dynamic patterns from massive traffic data. These multi-source data generally imply abundant potential dynamic patterns, including: 1) Spatial features. Traffic flow data own a significant relationship to the topological structure of the road networks. As illustrated in Fig.1, the nodes represent the roads, and links denote the connection relationship between roads. Besides, the traffic flow pattern changing from events in the upstream roads might also be propagated to the downstream roads in the following time intervals [5]. The connection relationship of road networks could also be changed, which increases the instability in modeling spatial features of traffic flow. 2) Temporal features. The current traffic flow is also strongly related to that in the neighboring time intervals since the dynamic events of traffic flow occur continuously over time [6], as shown by red arrows in Fig.1. Moreover, traffic flow patterns usually show different temporal periodicities, such as rush hour patterns, daily patterns, and weekly patterns. It is noteworthy that the interaction of temporal and spatial characteristics leads to the variability of traffic flow [7], as shown by blue dashed arrows in Fig.1. Therefore, the remaining crucial challenge is how to capture dynamic patterns from big traffic flow data and then predict traffic conditions accurately.

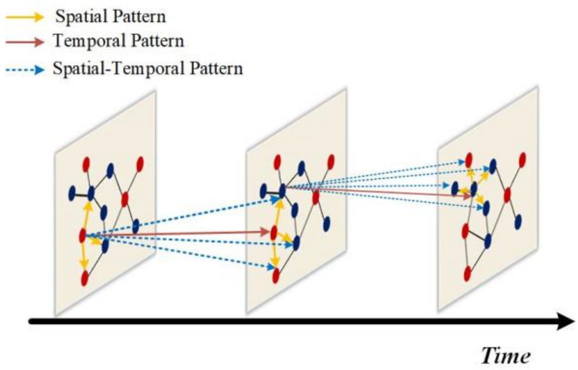


Fig.1: Spatial-temporal relationship of traffic flow in road networks.

In the past few years, several studies focusing on traffic flow prediction have been conducted. For example, the time series prediction methods such as the Auto-Regressive Integrated Moving Average (ARIMA) model [8] are the common approaches employed for short-term prediction tasks. However, it could only deal with essentially linear data. Furtherly, some non-linear models, including Support Vector Regression (SVR) [9], Bayesian Networks (BN) [10], and K-Nearest Neighbor (KNN) models [11], are more flexible for describing the non-linear relationship of the traffic flow. Still, they remain an evident challenge to discover intricate traffic patterns from big traffic data.

In recent years, several studies considered deep learning approaches as the best alternative to traffic prediction due to their ability to improve prediction performance. Some of these approaches mainly consider temporal features. For example, SBU- LSTM [12] is proposed based on Long Short-Term Memory (LSTM) [13], which takes into consideration forward and backward temporal features. However, considering only the temporal features is insufficient to satisfy the prediction performance requirement for complex traffic scenes. Furthermore, Convolutional Neural Networks (CNN) [14] and Time Series (TS) models were combined to capture the spatial and temporal patterns. In ST-3DNet [15], a 3D convolutional operator and a recalibration module were proposed to learn the spatial and temporal relations from different time terms. What's more, in ITRCN [16], CNN and Gated Recurrent Units (GRU) were integrated for learning spatial-temporal dependencies.

Although a considerable number of traditional and machine learning methods have been developed to improve the prediction accuracy of traffic flow, only a few of them consider the real topological relation of the road networks, which is pivotal for analyzing the propagation pattern of traffic congestion. To address this issue, the GCN was proposed to express the structural information of road networks and accurately predict traffic flow [17-19]. After that, STGCN [20] was developed based on GCN with a pure convolutional structure to extract road networks' temporal and spatial patterns simultaneously. Additionally, T-GCN [21] introduced a gated mechanism to the GCN layer to model the spatial-temporal relations with velocity features. STSGCN [7] used a module to capture local temporal and spatial features and stacks this module in the time dimension to capture global temporal features. Generally, these GCN-based methods could effectively express the spatial structure characteristics of traffic networks. Still, they show typically non-negligible deficient in modeling the dynamic changes of road networks.

Furtherly, GCN was also employed to build models for traffic flow prediction with dynamic characteristics of road networks. For example, ASTGCN [22] introduced the attention operators [23] and CNN to learn the spatial-temporal dependencies and then update the graph directly. With a unified model integrating LSTM and the attention mechanism, DGCN [24] was proposed to capture traffic features and get a better accuracy, which dynamically updates the Laplacian matrix and updates the whole graph indirectly. Although DGCN tried to improve the computation performance by sampling, it converted traffic flow characteristics into local features. Generally, it failed to keep the integrity of the temporal characteristics and aroused another hardness for updating the graph with global temporal attributes.

In general, there are still some crucial limitations of the state-of-the-art approach for traffic flow prediction. Firstly, the complete temporal features are ignored in model prediction performance because of the data sampling. Secondly, although both ASTGCN and DGCN considered the weekly and daily periodicities' traffic characteristics, they ignored the crucial neighbor period in feature construction and led to insufficient accuracy of traffic flow forecasting.

To address these drawbacks, with the inspiration of Chebyshev polynomial, we propose an improved dynamic Chebyshev graph convolution neural network model, called iDCGCN, to predict traffic flow accurately. Firstly, the iDCGCN model considers the neighboring time intervals of the input data to take advantage of the relevant time dimension. Secondly, the Laplacian matrix is dynamically updated with a multi-temporal convolution layer and a global spatial self-attention. The multi-temporal convolution layer is used to learn multi-scaled local temporal features from massive data. The global spatial self-attention module could also extract global spatial features, which remains complete temporal information to update the Laplacian matrix. Therefore, iDCGCN distinctively updates the Laplacian matrix with multi-local, global temporal characteristics and global spatial features.

What's more, iDCGCN owns the inherent ability to reduce the computational complexity for a larger traffic network. Besides, the Chebyshev graph convolution neural network layer aggregates the dynamic Laplacian matrix. The introduced temporal attention mechanism is also beneficial to learn more long-term traffic patterns. The experimental results show that iDCGCN outperforms the state-of-the-art forecasting models.

The main contributions of this work could be summarized as follows:

1) Unlike the existing methods using LSTM to update the Laplacian matrix, which usually leads to a considerable computation burden, inspired from the approximate estimation of convolution kernel by Chebyshev polynomial, a novel updating method for Laplacian matrix is proposed in iDCGCN based on the attention mechanism. This updating method approximately constructs features from different period data, and the experimental results show its superiority at the processing performance and accuracy.
2) Similar to Chebyshev polynomial, a novel feature construction method is proposed with multi-periodical representation, which integrates long-short temporal and local-global spatial features and could be employed for complicated feature representation.

3) Experiments were conducted on four actual traffic data sets, including PeMSD3, PeMSD4, PeMSD7, and PeMSD8, and results showed iDCGCN is superior to the baselines at both prediction accuracy and computation performance.

The rest of this article is organized as follows. In section 2, the road network definition and the problem description are introduced in detail; the construction process of the proposed model is mainly described in Section 3, and the experimental results and their discussion are provided to show the effectiveness of the proposed model in Section 4. Then we summarize our work and describe the direction of future work in Section 5.

2. Problem Description and Definition

2.1 Road Network Definition.

The road network topology relation is defined as an undirected graph $G = (V, E, A)$, where $V \in \mathbb{R}^N$ is a node-set used to describe roads; N denotes the total number of roads; E is an edge set indicating the connection relationship between nodes; $A \in \mathbb{R}^{N\times N}$ is the adjacent matrix, and $A_{i,j} = 1$ represents road i, and road j are connected and $A_{i,j} = 0$ shows no link between them. Another topological representation of road network is the Laplacian matrix, which could also be defined as $L = D - A$, where $D \in \mathbb{R}^{N\times N}$ is the degree matrix of A, $D_{i,i} = \sum_j^N A_{i,j}$ and D is a diagonal matrix.

2.2 Problem Description.

Traffic flow prediction intends to accurately predict the future road network traffic flow with the historical traffic data. The c^{th} traffic feature of road i in time interval t is defined as $x_t^{i,c}$, where $c \in \{1,2,3\}$ respectively denotes a specific feature in traffic feature set C = {traffic flow, road occupancy, average speed}. Specifically, $x_t^{i,1}$ denotes the traffic flow of road i in time slot t. $x_t^i = (x_t^{i,1}, x_t^{i,2}, x_t^{i,3}) \in \mathbb{R}^{|C|}$ stands for all the traffic features of road i in time interval t. $X_t =$

$(x_t^1, x_t^2, \cdots, x_t^i, \cdots, x_t^N) \in \mathbb{R}^{N\times|C|}$ contains all traffic features of all roads in time interval t, and $X_t^1 = (x_t^{1,1}, x_t^{2,1}, \cdots, x_t^{i,1}, \cdots, x_t^{N,1}) \in \mathbb{R}^N$ represents traffic flow of all roads in time slot t.

Then, given a road network G and historical traffic features of road networks over p time intervals $(X_1, X_2, \cdots, X_p)$. A mapping function f from the historical traffic data to its traffic flow in the flowing τ time intervals $(\tilde{X}_{p+1}^1, \tilde{X}_{p+2}^1, \cdots, \tilde{X}_{p+\tau}^1)$, as shown by the following equation:

$$(\tilde{X}_{p+1}^1, \tilde{X}_{p+2}^1, \cdots, \tilde{X}_{p+\tau}^1) = f((X_1, X_2, \cdots, X_p); G(V, E, A)) \quad (1)$$

3. Modeling and Methods

3.1 Model Framework.

To effectively learn dynamic spatial-temporal patterns from massive historical data, a novel method named iDCGCN is proposed in this work. The framework of iDCGCN mainly consists of three components: traffic data construction, two DST-Blocks (including a Laplacian matrix learning layer and a GCN-based forecasting model), and a feature fusion layer. The whole framework is demonstrated as Fig.2:

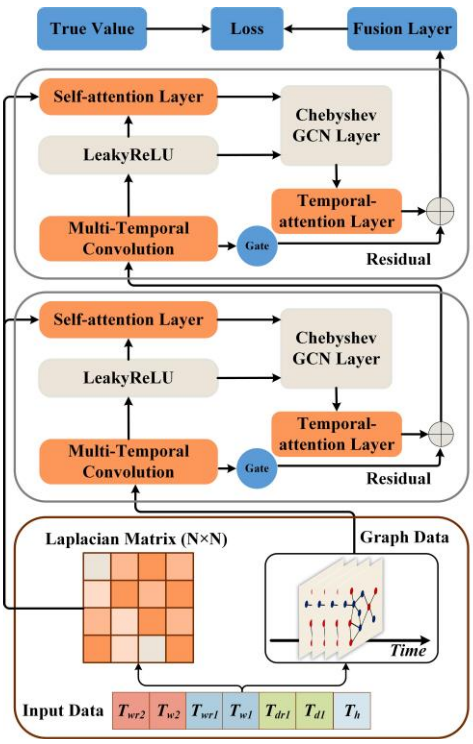


Fig.2: The framework of iDCGCN.

As shown in Fig.2, the Laplacian matrix and graph data are first constructed from raw input data. The graph data contains three types of periodicities, namely weekly-period, daily-period and recent. Secondly, a multi-temporal convolutional operator and a global spatial self-attention mechanism are integrated for learning multi-scaled temporal features and then generating the Laplacian matrix at the next moment. Then, upstream traffic features processed by multi-temporal convolution and the updated Laplacian matrix are fed into Chebyshev GCN to capture global spatial features. At the same time, the temporal attention layer is employed in dynamic global temporal characteristics. In DST-Block, a gate mechanism and a residual mechanism [25] are proposed to deliver upstream information to subsequent modules for processing selectively. Finally, a fusion layer is introduced to merge the output vector of DST-Block for the loss evaluation at the last stage.

3.2 Traffic Data Construction.

Considerable studies have shown that traffic flow owns strong relation between the adjacent time intervals [6]. Meanwhile, it also exhibits evitable temporal periodicities. Specifically, consecutive workdays might have similar patterns, and traffic flow may be close on the same day as neighborhood weeks. Several studies [22,24,26] integrated the patterns of weekly-period, daily-period, and recent-period traffic features for traffic flow prediction. In these studies, the weekly-period segment corresponds to the same forecasting period in the several neighboring weeks; the daily-period segment corresponds to the same prediction period in the several adjacent days, and the recent-period is the neighboring time intervals of the prediction periods. However, the pattern of the recent period could be generalized to every day. For instance, the traffic flow tendency from 8 AM to 9 AM on Monday may be inferred from 7 AM to 8 AM on the previous Monday.

As shown in Fig.3, we propose a novel method to construct the weekly-periodic segment T_w and daily-periodic segment T_d by adding the former neighboring time intervals T_{wr2}, T_{wr1} and T_{dr1} of the original segments T_{w2}, $T_{w1}$and T_{d1} respectively, which is different from the existing ways of traffic data construction.

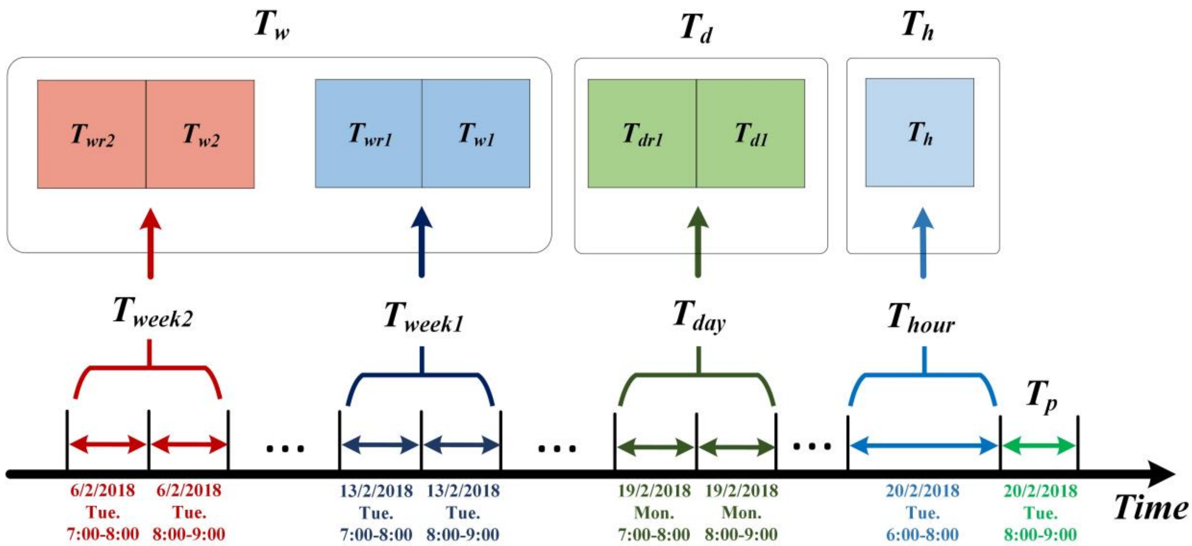


Fig.3: An example of traffic feature reconstruction in the temporal direction. Suppose the predicting period $|T_p|$ is 1 hour; recent period $|T_h|$ is 2 hours and day period $|T_{d1}|$ and $|T_{d1}|$'s neighboring time intervals $|T_{dr1}|$ are both 1 hour. Week period $|T_{w1}|$ and $|T_{w1}|$'s neighboring time intervals $|T_{wr1}|$ are both 1 hour. Similarly, $|T_{w2}|$ and $|T_{wr2}|$ are 1 hour. $|\cdot|$ indicates the length of the corresponding time dimension.

To eliminate the dimensional difference between several traffic properties, we normalized the T_w, T_d and T_h respectively using the following equations.

$$x' = \frac{x - mean(x)}{std(x)} \tag{2}$$

Where $mean()$ is used to calculate the mean; $std()$ is used to calculate the standard deviation and x stand for T_w, T_d and T_h respectively.

The standardized traffic data of $T_w \in \mathbb{R}^{|C|\times N\times|T_w|}$, $T_d \in \mathbb{R}^{|C|\times N\times|T_d|}$ and $T_h \in \mathbb{R}^{|C|\times N\times|T_h|}$ are concatenated as input data X of the Laplacian matrix learning layer, where $X \in \mathbb{R}^{|C|\times N\times T}$, $T = |T_w| + |T_d| + |T_h|$, and $|T_w|$, $|T_d|$, and $|T_h|$ represent the time length of T_w, T_d, and T_h. The initial number of features is represented as $|\mathrm{C}|$, and N represents the number of roads.

3.3 Laplacian Matrix Learning Layer.

A learning layer is proposed to adaptively update the Laplacian matrix from dynamic multi-temporal features and global spatial characteristics to calculate reasonably the Laplacian matrix. In this layer, multi-temporal convolutional kernels are applied to extract multi-local temporal information on each road. Meanwhile, a global spatial self-attention operator is employed to learn the spatial relevance between roads with the complete global temporal features for updating the Laplacian matrix dynamically.

3.3.1 Multi-Temporal Convolution Layer.

A multi-temporal convolution layer is designed to extract the multi-local features in the time dimension. The layer utilizes multiple 2-D convolution kernels to capture the local temporal information of different time ranges from the reconstructed traffic data. Each temporal convolution kernel is a sliding window of size 1×s in the temporal direction. In this method, the traffic features are combined in the temporal dimension, as illustrated in Fig.4.

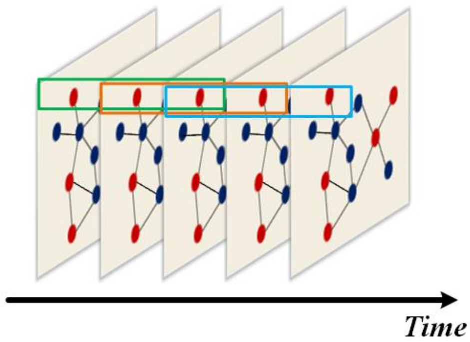


Fig.4: An example of temporal convolution with $s = 3$.

The temporal convolution could be represented in eq. 3 and 4 respectively as follows:

$$Conv_{1\times s}(X) = W_{TC} \otimes X + b_{TC} \tag{3}$$

$$X_{TC} = LeakyReLU(Conv_{1\times s}(X)) \tag{4}$$

Where $\otimes$ stands for the convolution operation; W_{TC} denotes one of the multi-temporal convolution kernels whose size is $1 \times s$; b_{TC} is the learnable bias; $X \in \mathbb{R}^{|C| \times N \times T}$ represents the input of the temporal convolution layer; LeakyReLU is used as the activation function, and $X_{TC} \in \mathbb{R}^{F \times N \times T}$ with F as the number of new feature dimensions.

After multi-temporal convolution operations, X_{TC} is obtained and then processed by a gate whose output X_{res} is regarded as a residual for retaining valuable information of the previous network. The definition is demonstrated in eq.5 and 6, respectively, as follows.

$$G_{\text{gate}} = \delta(Conv_{1 \times s}(\mathrm{X})) \tag{5}$$

$$\mathrm{X}_{res} = G_{\text{gate}} \odot X_{TC} \tag{6}$$

Where $G_{\text{gate}} \in \mathbb{R}^{F \times N \times T}$; σ denotes the sigmoid function; $\mathrm{X}_{res} \in \mathbb{R}^{F \times N \times T}$, and $\odot$ represents Hadamard product. The pseudo-code of iDCGCN is summarized as follows:

Algorithm: iDCGCN

Input: Construct ① and ② from the original data:
① $X \in \mathbb{R}^{|C| \times N \times T}$ concatenated by $T_w \in \mathbb{R}^{|C| \times N \times |T_w|}$, $T_d \in \mathbb{R}^{|C| \times N \times |T_d|}$, and $T_h \in \mathbb{R}^{|C| \times N \times |T_h|}$;
② Normalized Laplacian matrix $\tilde{L} \in \mathbb{R}^{N \times N}$;
for i^{th} DST-Block, $i = 1, 2, \ldots, n$ **do**
1. Get $X_{TC} \in \mathbb{R}^{F \times N \times T}$ and $G_{\text{gate}} \in \mathbb{R}^{F \times N \times T}$ by (3) (4) (5) from ①;
2. Get residual $\mathrm{X}_{res} \in \mathbb{R}^{F \times N \times T}$ by (6);
3. Get spatial-temporal coefficient matrix $S_a^{'} \in \mathbb{R}^{N \times N}$ by (7) (8) (9);
4. Update Laplacian matrix $\tilde{L}_u \in \mathbb{R}^{N \times N}$ by (11) from ②;
5. Update Chebyshev Graph Convolution Network $X_{GCN} \in \mathbb{R}^{F \times N \times T}$ by (12);
6. Get temporal similarity coefficient matrix $\mathrm{E}^{'} \in \mathbb{R}^{T \times T}$ by (13) (14);
7. Get $X_{TA} \in \mathbb{R}^{F \times N \times T}$ by (15);
8. $X_{out} = X_{TA} + \mathrm{X}_{res}$ by (16);

end for
Fusion layer: $(\tilde{X}_{p+1}^{1}, \tilde{X}_{p+2}^{1}, \cdots, \tilde{X}_{p+\tau}^{1}) = \sum_{m=1}^{T/\tau} \omega_m \odot X_{outm}^{'}$ by (17).
Output: $(\tilde{X}_{p+1}^{1}, \tilde{X}_{p+2}^{1}, \cdots, \tilde{X}_{p+\tau}^{1})$

3.3.2 Global Spatial Self-attention.

Previous studies in the literature usually used topological matrices, such as k-hop adjacency matrix [27] or k-hop Laplacian matrix [28], to describe road networks' fixed k-hop connection relationship. These "hard-link" methods consider the locality of road connections solely and ignore the potential global relationship between the roads. The missing information about roads' association is generally a crucial limitation regarding the Laplacian matrix. For example, DGCN omits global temporal features while updating the Laplacian matrix. To address this issue, we introduced a global spatial self-attention layer [29] to dynamically learn the spatial relationship of the road network in every time interval, as shown in the following equations:

$$Att = QK^{Tr} \tag{7}$$

$$Att'_{i,j} = \frac{exp(Att_{i,j})}{\sum_{j=1}^{N} exp(Att_{i,j})} \tag{8}$$

$$S'_a = \sigma(Att'V) \tag{9}$$

Where, $Q,\ K,\ V \in \mathbb{R}^{N\times N}$ represent the query, key, and value respectively transformed from the X_{TC}, $Q = Conv_{1\times1}(X_{TC})W_Q$, $K = Conv_{1\times1}(X_{TC})W_K$, and $V = Conv_{1\times1}(X_{TC})W_V$. $Conv_{1\times1}(X_{TC}) \in \mathbb{R}^{N\times T}$. $W_Q,\ W_K,\ W_V \in \mathbb{R}^{T\times N}$ are learnable weight matrixes, Tr stands for the transposition operation. Att measures the similarity of Q and K. $Att'_{i,j}$ is calculated by the softmax function, which expresses the coupling strength between road i and j. $S'_a \in \mathbb{R}^{N\times N}$ denotes the spatial-temporal coefficient matrix. In addition, the spatial-temporal coefficient matrix S'_a is directly applied in the Laplacian matrix, which adjusts the spatial-temporal correlations between more roads appropriately.

3.4 GCN-based Traffic Flow Prediction.

The dynamic Laplacian matrix is transferred into the Chebyshev graph convolution network and updates the entire graph indirectly. Additionally, the temporal attention operator is employed to learn the temporal relativity over a long-time range.

3.4.1 Dynamic Chebyshev Graph Convolution Network.

In this work, a dynamic Chebyshev graph convolution network is updated with a dynamic Laplacian matrix. Given a graph G and the graph data x, a Chebyshev graph convolution operator is defined as follows:

$$g_\theta * \mathrm{G}\, x = g_\theta(\tilde{L})x = \sum_{k=0}^{K-1} \theta_k\, T_k(\tilde{L})x \tag{10}$$

where, g_θ represents a trainable kernel; $*\,\mathrm{G}$ represents graph convolutional operation; $L = D - A \in \mathbb{R}^{N\times N}$ is the Laplacian matrix. $\theta_k \in \mathbb{R}^K$ denotes a parameter vector of Chebyshev coefficients, and K is the order of the Chebyshev polynomials. $\tilde{L} = \frac{2}{\lambda_{max}} L - I \in \mathbb{R}^{N\times N}$ represents a scaled eigenvalues matrix whose values lie in [-1,1], λ_{max} is the largest eigenvalue of L, $I \in \mathbb{R}^{N\times N}$ is a unit matrix; $T_k(\tilde{L})$ is a Chebyshev polynomial function calculated recursively from $\tilde{L}$: $T_k(\tilde{L}) = 2\tilde{L}T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L}), T_0(\tilde{L}) = I, T_1(\tilde{L}) = \tilde{L}$.

In this work, the updated Laplacian matrix could be obtained from the following equation.

$$\tilde{L}_u = \tilde{L} \odot S'_a \tag{11}$$

The output of the multi-temporal convolution layer with a gate mechanism is considered as the input data of Chebyshev GCN, as shown in Fig.2. It integrates the multi-local temporal relations of traffic graph data and retains most temporal features over the whole-time range of traffic data. Therefore, the Chebyshev GCN is defined as follows:

$$X_{GCN} = \sigma(\sum_{k=0}^{K-1} \theta_k\, T_k(\tilde{L}_u) X_{TC}) \tag{12}$$

where $X_{GCN} \in \mathbb{R}^{F\times N\times T}$ is the output of Chebyshev GCN; $X_{TC} \in \mathbb{R}^{F\times N\times T}$ denotes the input data of Chebyshev GCN, $T_k(\tilde{L}_u) \in \mathbb{R}^{N\times N}$.

3.4.2 Temporal Attention.

Previously, the multi-temporal convolutional kernels are utilized to learn the short-term pattern. Furtherly, to learn the long-term traffic features for improving the traffic flow forecasting accuracy, a temporal attention mechanism [22] is adopted to explore the long-term pattern. The calculation of temporal similarity coefficient matrix E' is represented in the following equations:

$$E = V_e \cdot \sigma(((X_{GCN})^{Tr}\mathrm{W}_1)\mathrm{W}_2(X_{GCN}\mathrm{W}_3) + b_e) \tag{13}$$

$$E'_{i,j} = \frac{e^{\mathrm{E}_{i,j}}}{\sum_{j=1}^{T} e^{\mathrm{E}_{i,j}}} \tag{14}$$

where $V_e \in \mathbb{R}^{T\times T}$, $\mathrm{W}_1 \in \mathbb{R}^{F}$, $\mathrm{W}_2 \in \mathbb{R}^{N\times F}$, $\mathrm{W}_3 \in \mathbb{R}^{N}$ and $b_e \in \mathbb{R}^{T\times T}$ are trainable parameters; $E \in \mathbb{R}^{T\times T}$ is processed through the normalization of the softmax function; $E'_{i,j}$ is the relevance between time moment i and j. The temporal similarity coefficient matrix is applied to the output of Chebyshev GCN as demonstrated in the following equations:

$$X_{TA} = LeakyReLU(X_{GCN}E') \tag{15}$$

$$X_{out} = X_{TA} + \mathrm{X}_{res} \tag{16}$$

where $X_{TA} \in \mathbb{R}^{F\times N\times T}$, $E' \in \mathbb{R}^{T\times T}$, $X_{out} \in \mathbb{R}^{F\times N\times T}$.

3.4.3 Traffic Feature Fusion.

To predict traffic flow with different temporal lengths, we design a traffic feature fusion layer to increase the diversity of feature extraction. Firstly, we separate X_{out} into $\frac{T}{\tau}$ segments in the temporal dimension and obtain a new set $X'_{out} = \{X_{out1}, X_{out2}, \cdots, X_{outm}, \cdots, X_{out\frac{T}{\tau}}\}$, where τ is the length of predicting period, and $X_{outm} \in \mathbb{R}^{F\times N\times \tau}$. Later, a convolutional kernel of size 1×1 is introduced to integrate information of F channels in each element, $X'_{outm} = Conv_{1\times 1}(X_{outm}) \in \mathbb{R}^{N\times \tau}$. Finally, all the elements weighted are fused as follows:

$$(\tilde{X}^1_{p+1}, \tilde{X}^1_{p+2}, \cdots, \tilde{X}^1_{p+\tau}) = \sum_{m=1}^{T/\tau} \omega_m \odot X'_{outm} \tag{17}$$

where $(\tilde{X}_{p+1}^{1}, \tilde{X}_{p+2}^{1}, \cdots, \tilde{X}_{p+\tau}^{1}) \in \mathbb{R}^{N\times\tau}$ represents the predicting results; $\omega_m \in \mathbb{R}^{N\times\tau}$ denotes the trainable weight.

In addition, the Mean Square Error (MSE) is employed as a loss function to evaluate the difference between real traffic flow and predicted traffic flow:

$$loss = \frac{\sum_{i=p+1}^{p+\tau}\sum_{j=1}^{N}(x_i^{j,1} - \tilde{x}_i^{j,1})^2}{\tau \cdot N} \tag{18}$$

where, $x_i^{j,1}$ represents the ground truth, and $\tilde{x}_i^{j,1}$ denotes the value of prediction.

4. Results and Discussion

This section uses four real traffic datasets to evaluate traffic flow prediction performance with the iDCGCN approach.

4.1 Experimental Settings.

The experimental settings are introduced in this section, including data description, parameter settings, comparison methods, and evaluation metrics.

4.1.1 Data Description.

As shown in Table 1, the experiments are conducted on four real highway datasets: PeMSD3, PeMSD4, PeMSD7, and PeMSD8, gathered from the Caltrans Performance Measurement System (PeMS) [30] and sampled every 5 minutes. PeMSD4 and PeMSD8 contain three traffic features: traffic flow, road occupancy, and average speed. PeMSD3 and PeMSD7 contain the characteristic of traffic flow. PeMSD4 was collected by 307 detectors in the San Francisco Bay Area spans from January to February in 2018, and PeMSD8 was collected by 170 detectors in the San Bernardino area spans from July to August in 2016. PeMSD3 and PeMSD7 contain 358 and 883 detectors respectively. As stated in Section 3.2, both datasets are processed. The training set, validation set, and testing set are generated at a ratio of 3:1:1. Each piece of data is a combination of T_w, T_d, and T_h illustrated in Fig.3, where the difference is the start time of the forecast.

Table 1. Description of datasets

Datasets	Amount	Nodes	Features	Time interval
PeMSD3	26208	358	1	9/1/2018-11/30/2018
PeMSD4	16992	307	3	1/12018-2/28/2018
PeMSD7	28224	883	1	5/1/2017-8/31/2017
PeMSD8	17856	170	3	7/1/2016-8/31/2016

4.1.2 Parameter Settings.

iDCGCN is implemented by the Pytorch framework of version 1.7.1, and the experimental equipment includes an Intel Core i7-10700KF processor and a Graphics card of NVIDIA GeForce GTX 1080Ti with 11GB memory. The predicting period is set as $\tau = 12$(1 hour), and the lengths of recent-period, daily-period, and weekly-period are set as $T_w = 48$, $T_d = 24$, and $T_h = 24$. Multi-temporal convolution kernels are 64, and the kernel sizes are set to 1, 3, and 5, respectively. The number of DST-block is 2. In Chebyshev GCN, the approximation level K of Chebyshev polynomial is set as 3, which is the same as ASTGCN [22]. The training epochs are set to 40. The size of each batch is 16, and we set the learning rate to 0.0005. The MSE is adopted as a loss function, and Adam optimizer is used for backpropagation.

4.1.3 Comparison Methods and Evaluation Metrics.

The proposed traffic flow prediction method is compared with seven forecasting benchmark models, including the Historical Average method (HA), ARIMA, LSTM, GRU, STGCN, ASTGCN, DGCN. On these methods, we use historical traffic data in consecutive 12-time intervals for the former four methods to predict the traffic flow of 12-time intervals. In GCN-based methods, three temporal segments are respectively set as: $T_w = 24$, $T_d = 12$, $T_h = 24$, and $\tau = 12$.

We utilize three evaluation metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), to evaluate the prediction performance of iDCGCN and compare performance results. These metrics are defined as follows:

$$MAE = \frac{\sum_{i=p+1}^{p+\tau}\sum_{j=1}^{N}|(x_i^{j,1}-\tilde{x}_i^{j,1})|}{\tau\cdot N} \tag{19}$$

$$RMSE = \sqrt{\frac{\sum_{i=p+1}^{p+\tau}\sum_{j=1}^{N}(x_i^{j,1}-\tilde{x}_i^{j,1})^2}{\tau\cdot N}} \tag{20}$$

$$\mathrm{MAPE} = \frac{1}{\tau\cdot N}\sum_{i=p+1}^{p+\tau}\sum_{j=1}^{N}\left|\frac{x_i^{j,1}-\tilde{x}_i^{j,1}}{x_i^{j,1}}\right| \tag{21}$$

4.2 Results Analysis and Discussion.

This section compared the short-period and long-period prediction performance between iDCGCN and the benchmark forecasting methods. The experimental results were compared based on PeMSD4 and PeMSD8 using different methods for forecasting traffic flow in one hour ($\tau = 12$). The experimental results of our method and the benchmark models using two real highway datasets are demonstrated in Table 2.

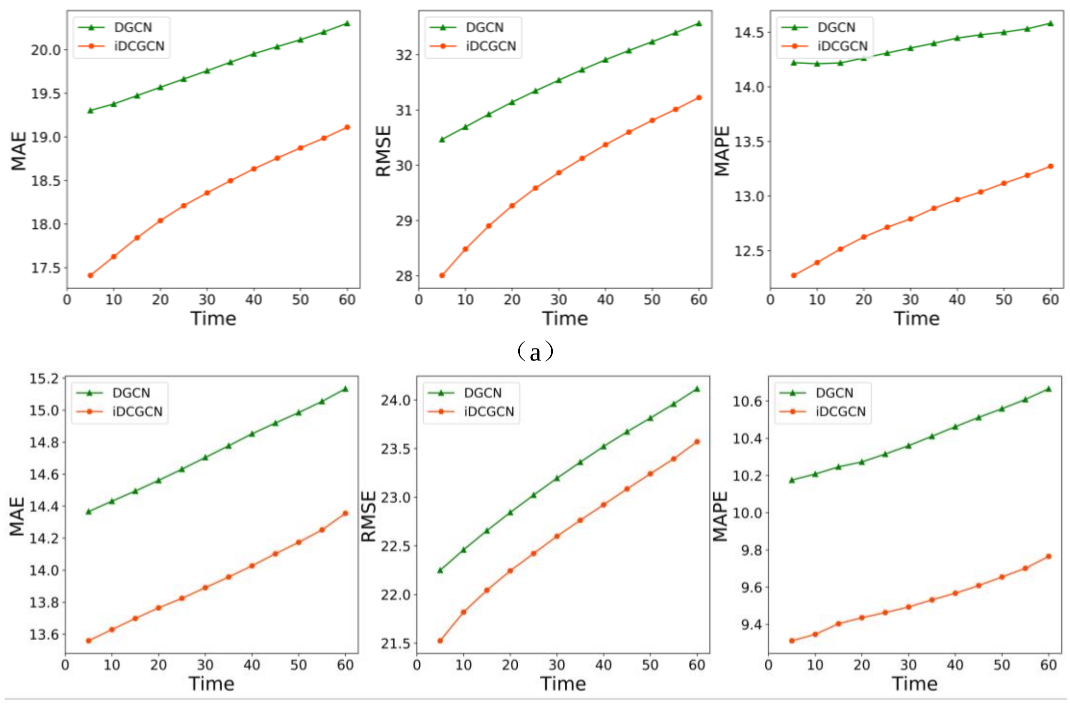
Table 2. Experimental results of multiple methods on forecasting traffic flow in 1 hour

Model	PeMSD4			PeMSD8		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	66.48	51.32	72.23	54.58	42.36	30.12
ARIMA	77.38	53.00	54.23	45.81	42.86	35.49
LSTM	43.43	28.85	25.22	35.31	23.01	19.61
GRU	44.02	29.41	24.70	36.33	23.68	18.99
STGCN	41.08	27.62	24.05	32.22	21.36	15.92
STSGCN	38.56	24.62	16.41	30.64	20.12	13.04
ASTGCN	34.25	21.49	15.50	27.09	17.87	12.21
DGCN	31.53	19.67	14.3	23.92	15.18	10.31
iDCGCN_D (ours)	31.54	19.60	13.61	23.52	14.78	10.25
iDCGCN (ours)	**31.31**	**19.5**	**13.58**	**23.19**	**14.16**	**9.44**

As illustrated in Table 1, the proposed iDCGCN is significantly superior to the benchmark methods. The values of T_{wr2}, T_{wr1}, and T_{dr1} in the iDCGCN_D model are 0. The conventional time series forecasting methods HA and ARIMA achieve deficient performance because they can only capture linear regularity and do not consider the road network topology. The prediction precision of LSTM and GRU is improved compared to the former two models because they can effectively extract temporal information from non-linear time series data.

The GCN-based forecasting models, including STGCN, STSGCN, ASTGCN, and DGCN, generally achieve an obvious improvement compared to time series methods because they fully consider the connection relationship between roads. For instance, STGCN outperforms LSTM because it combines GCN with gated linear units to learn traffic data's spatial and temporal characteristics. STSGCN captures local temporal-spatial characteristics and expands along the time dimension. ASTGCN considers three traffic patterns using the attention mechanism and produces improved performance results. Furthermore, DGCN made a further improvement to ASTGCN by indirectly updating the graph network.

In addition, we compare the traffic flow prediction performance of iDCGCN and DGCN on 12 different predicting time intervals to evaluate the short-period and long-period prediction performance. As shown in Fig.5, the proposed iDCGCN achieves better forecasting results in all predicting time intervals on both PeMSD4 and PeMSD8. As the length of forecasting interval increases, the growth rates of RMSE, MAE, and MAPE of iDCGCN are getting lower, which shows that iDCGCN is beneficial to simultaneously capture the spatial and temporal information of short-term and long-term patterns.



(a)

(b)

Fig.5: The forecasting results of models on different predicting time intervals. (a) RMSE, MAE, and MAPE of two methods on PeMSD4. (b) RMSE, MAE, and MAPE of two methods on PeMSD8.

Generally, the model performance evaluation should consider both the forecasting accuracy and time consumption. We compared our model's prediction performance with the latest DGCN model to verify the proposed model's effectiveness, as shown in Table 3. It's noteworthy that the experimental results show that the used memory of the DGCN model increases sharply while the number of data increases. As a result, dataset PeMSD7 could not be run by DGCN for the out-of-memory, which is denoted as **NULL** in the following table.

Table 3. Effectiveness and efficiency of the model

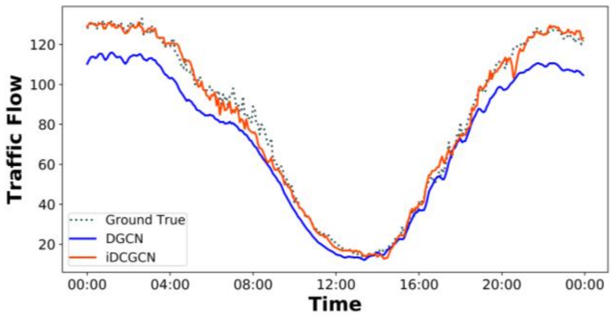
Models	**DGCN/iDCGCN**			
Datasets	**PeMSD3 (358)**	**PeMSD4 (307)**	**PeMSD7 (883)**	**PeMSD8(170)**
RMSE	34.76/30.57	31.53/31.31	**NULL**/34.02	23.92/23.19
MAE	19.63/17.84	19.67/19.5	**NULL**/21.32	15.18/14.16
MAPE	19.43/18.16	14.3/13.58	**NULL**/9.88	10.31/9.44
Training time (s)	578.20/274.77	253.21/199.04	**NULL**/1039.23	92.58/78.14
Testing time (s)	59.75/19.19	27.06/13.24	**NULL**/76.21	9.71/5.27

To confirm the effect of each component in the model, we take out one part and test the prediction accuracy. The prediction results after eliminating one component are demonstrated in Table 4, where iDCGCN_Att, iDCGCN_Gate, and iDCGCN_Conv represent the model after removing the temporal attention module, gate module, and multi-temporal convolution layer, respectively.

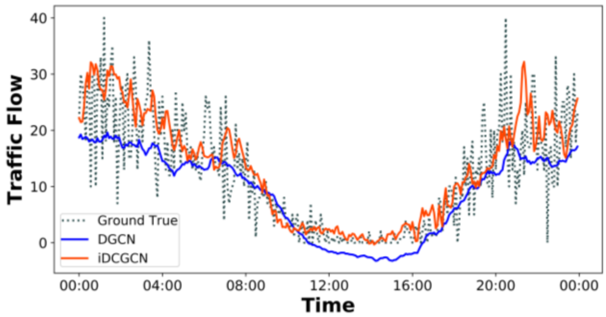
Table 4. The impact of different components in iDCGCN

Model	PeMSD4			PeMSD8		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
iDCGCN_Att	31.54	19.32	13.64	23.89	14.53	9.88
iDCGCN_Gate	32.02	19.63	13.64	23.21	14.32	9.45
iDCGCN_Conv	31.85	19.63	13.67	23.25	14.42	10.08
iDCGCN	**31.31**	**19.5**	**13.58**	**23.19**	**14.16**	**9.44**

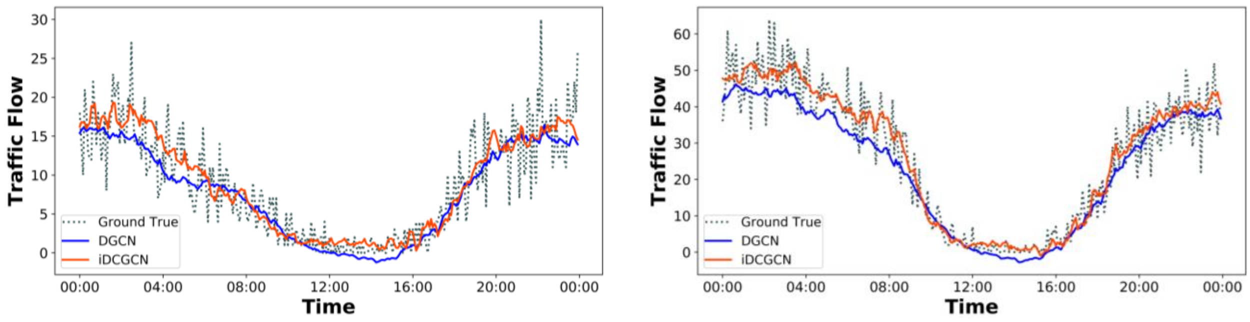
To further validate the prediction performance of the proposed iDCGCN, we randomly chose four roads (Road 147, 154, 155, and 169) in the PeMSD8 datasets and visualized the prediction results of DGCN and iDCGCN, as shown in Fig.6. The comparison results show that the proposed method could capture the changing trend of traffic flow accurately. For example, Fig.6 (a) shows that the prediction values of DGCN are lower than the actual values, while the forecasting results of iDCGCN are significantly closer to the actual values. Fig.6 (b-d) shows that iDCGCN could better fit the serration of traffic flow when the actual traffic flow data changes and fluctuates evidently.



(a)



(b)



(c) (d)

Fig.6: The forecasting displays of iDCGCN and DGCN of different roads on August 18, 2016.

5. Conclusions

Traffic flow prediction is essential to the ITS and its related applications. However, learning its patterns from spatial and temporal information on sophisticated historical traffic data is still a significant challenge. In this work, from the inspiration of Chebyshev polynomial, we propose a novel dynamic Chebyshev graph convolution network model, named iDCGCN, which approximately constructs features from different period data with an attention mechanism. This approximate method is beneficial for reducing the model's time complexity and capturing the more long-term features. Furthermore, a novel feature construction method is proposed in iDCGCN with multi-periodical representation, which integrates long-short temporal and local-global spatial features for complicated feature representation. Abundant experiments were conducted on four traffic data sets, including PeMSD3, PeMSD4, PeMSD7, and PeMSD8. Results showed iDCGCN outperforms the baselines at both predictions accuracy and computation performance.

In the future work, we will concentrate on further improving its adaptability by considering the external influence factors of traffic flow, such as weather conditions, holidays, traffic accidents, etc. Future studies will also consider how to balance training performance, reliability, toughness, economic efficiency, and ease of use.

Data Availability

The original highway datasets are derived from [22], and further raw data used in this work can be obtained from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest.

Acknowledgments

This work is supported in part by projects of the National Natural Science Foundation of China (41971340, 41471333, 61304199) and Fujian Provincial Department of Science and Technology (2021Y4019, 2020D002, 2020L3014, 2019I0019).

References

[1].Tu, Y., Lin, S., Qiao, J.&Liu, B. J. A. I., Deep traffic congestion prediction model based on road segment grouping, Applied Intelligence, 1-23, (2021).
[2].Belhadi, A., Djenouri, Y., Djenouri, D.&Lin, J. C.-W. J. A. I., A recurrent neural network for urban long-term traffic flow forecasting, Applied Intelligence, 50, 3252-3265, (2020).
[3].Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X.&Chen, C., Data-driven intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems, 12, 1624-1639, (2011).
[4].Li, W., Wang, X., Zhang, Y.&Wu, Q., Traffic flow prediction over muti-sensor data correlation with graph convolution network, Neurocomputing, 427, 50-63, (2021).
[5].Nguyen, H., Liu, W.&Chen, F., Discovering congestion propagation patterns in spatio-temporal traffic data, IEEE Transactions on Big Data, 3, 169-180, (2016).
[6].Zhao, Z., Chen, W., Wu, X., Chen, P. C.&Liu, J., LSTM network: a deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems, 11, 68-75, (2017).
[7].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. 2020, 34, 914-921.
[8].Kumar, S. V.&Vanajakshi, L., Short-term traffic flow prediction using seasonal ARIMA model with limited input data, European Transport Research Review, 7, 1-9, (2015).
[9].Wang, J.&Shi, Q., Short-term traffic speed forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory, Transportation Research Part C: Emerging Technologies, 27, 219-232, (2013).
[10].Kim, J.&Wang, G., Diagnosis and prediction of traffic congestion on urban road networks using Bayesian networks, Transportation Research Record, 2595, 108-118, (2016).
[11].Sun, B., Cheng, W., Goswami, P.&Bai, G., Short-term traffic forecasting using self-adjusting k-nearest neighbours, IET Intelligent Transport Systems, 12, 41-48, (2017).
[12].Cui, Z., Ke, R., Pu, Z.&Wang, Y., Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, arXiv preprint arXiv:1801.02143, (2018).
[13].Hochreiter, S.&Schmidhuber, J., Long short-term memory, Neural computation, 9, 1735-1780, (1997).
[14].Maheshan, M., Harish, B.&Nagadarshan, N., A Convolution Neural Network Engine for Sclera Recognition, International Journal of Interactive Multimedia & Artificial Intelligence, 6, (2020).
[15].Guo, S., Lin, Y., Li, S., Chen, Z.&Wan, H., Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting, IEEE Transactions on Intelligent Transportation Systems, 20, 3913-3926, (2019).
[16].Cao, X., Zhong, Y., Zhou, Y., Wang, J., Zhu, C.&Zhang, W., Interactive temporal recurrent convolution network for traffic prediction in data centers, IEEE Access, 6, 5276-5289, (2017).
[17].Bruna, J., Zaremba, W., Szlam, A.&LeCun, Y., Spectral networks and locally connected networks on graphs, arXiv preprint arXiv:1312.6203, (2013).

[18].Defferrard, M., Bresson, X.&Vandergheynst, P., Convolutional neural networks on graphs with fast localized spectral filtering, arXiv preprint arXiv:1606.09375, (2016).
[19].Kipf, T. N.&Welling, M., Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907, (2016).
[20].Yu, B., Yin, H.&Zhu, Z., Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, arXiv preprint arXiv:1709.04875, (2017).
[21].Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M.&Li, H., T-gcn: A temporal graph convolutional network for traffic prediction, IEEE Transactions on Intelligent Transportation Systems, 21, 3848-3858, (2019).
[22].Guo, S., Lin, Y., Feng, N., Song, C.&Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33, 922-929.
[23].Feng, X., Guo, J., Qin, B., Liu, T.&Liu, Y. Effective Deep Memory Networks for Distant Supervised Relation Extraction. In: IJCAI. 2017, 4002-4008.
[24].Guo, K., Hu, Y., Qian, Z., Sun, Y., Gao, J.&Yin, B., Dynamic Graph Convolution Network for Traffic Forecasting Based on Latent Network of Laplace Matrix Estimation, IEEE Transactions on Intelligent Transportation Systems, (2020).
[25].He, K., Zhang, X., Ren, S.&Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, 770-778.
[26].Chen, X., Zhang, Y., Du, L., Fang, Z., Ren, Y., Bian, K.&Xie, K. J. a. p. a., TSSRGCN: Temporal Spectral Spatial Retrieval Graph Convolutional Network for Traffic Flow Forecasting, arXiv preprint arXiv:2011.14638, (2020).
[27].Zhang, Z., Li, M., Lin, X., Wang, Y.&He, F., Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies, Transportation research part C: emerging technologies, 105, 297-322, (2019).
[28].Diao, Z., Wang, X., Zhang, D., Liu, Y., Xie, K.&He, S. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33, 890-897.
[29].Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L.&Polosukhin, I. J. a. p. a., Attention is all you need, arXiv preprint arXiv:1706.03762, (2017).
[30].Chen, C., Petty, K., Skabardonis, A., Varaiya, P.&Jia, Z., Freeway performance measurement system: mining loop detector data, Transportation Research Record, 1748, 96-102, (2001).