Spatial–Temporal Chebyshev Graph Neural Network for Traffic Flow Prediction in IoT-Based ITS

Biwei Yan, Guijuan Wang, Jiguo Yu, *Senior Member, IEEE*, Xiaozheng Jin, *Member, IEEE*, and Hongliang Zhang

***Abstract*—As one of the most widely used applications of the Internet of Things (IoT), intelligent transportation system (ITS) is of great significance for urban traffic planning, traffic control, and traffic guidance. However, widespread traffic congestion occurs with the increased number of vehicles. The traffic flow prediction is a good idea for traffic congestion. Therefore, many schemes have been proposed for accurate and real-time traffic flow prediction, but there still exist many issues, including low accuracy, weak adaptability and inferior real-time. Meanwhile, the complex spatial and temporal dependencies in traffic flow are still challenging. To address the above issues, we propose a novel spatial-temporal Chebyshev graph neural network model (ST-ChebNet) for traffic flow prediction to capture the spatial-temporal features, which can ensure accurate traffic flow prediction. Concretely, we first add a fully connected layer to fuse the features of traffic data into a new feature to generate a matrix, and then the long short-term memory (LSTM) model is adopted to learn traffic state changes for capturing the temporal dependencies. Then, we use the Chebyshev graph neural network (ChebNet) to learn the complex topological structures in the traffic network for capturing the spatial dependencies. Eventually, the spatial features and the temporal features are fused to guarantee the traffic flow prediction. The experiments show that ST-ChebNet can make accurate and real-time traffic flow prediction compared with other eight baseline methods on real-world traffic data sets PeMS.**

***Index Terms*—Graph neural network (GNN), intelligent transportation system (ITS), long short-term memory (LSTM), spatial-temporal graph data, traffic flow prediction.**

Manuscript received April 1, 2021; revised June 1, 2021 and July 8, 2021; accepted August 2, 2021. Date of publication August 17, 2021; date of current version June 7, 2022. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61832012, Grant 61771289, and Grant 61672321; in part by the Key Research and Development Program of Shandong Province under Grant 2019JZZY020124; and in part by the Pilot Project for Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2020KJC-ZD02. *(Corresponding author: Guijuan Wang.)*

Biwei Yan, Guijuan Wang, and Hongliang Zhang are with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, Shandong, China (e-mail: for_yanbiwei@163.com; guijuan_wang@126.com; hongliangzhang2021@163.com).

Jiguo Yu and Xiaozheng Jin are with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, Shandong, China, and also with the Shandong Laboratory of Computer Networks, Jinan 250014, China (e-mail: jiguoyu@sina.com; jin445118@163.com).

Digital Object Identifier 10.1109/JIOT.2021.3105446

I. Introduction

THE PROMINENT development of the Internet of Things (IoT), 5G, and artificial intelligence (AI) has produced lots of IoT application systems. Among them, the intelligent transportation system (ITS) is one of the most promising applications of IoT, which has provided unprecedentedly services [1]. ITS integrates the IoT, cloud computing, and mobile Internet to further improve the traffic efficiency and management level. In the ITS, the IoT can fully perceive the traffic infrastructure and the traffic management department can fully mine and use traffic information, guide vehicles correctly and ensure traffic security. However, with the acceleration of urbanization, the number of vehicles in cities has increased rapidly, resulting in increasingly widespread traffic congestion. The inefficiencies, time delays, and environmental issues caused by traffic congestion are too many to list, and there are more and more "road rage" and "passion crimes" caused by traffic congestion, which are the main features of modern "urban diseases." Therefore, improving traffic conditions is essential for enhancing city efficiency, reducing carbon emissions, ameliorating the economy, and improving people's quality of life. Traffic flow prediction can predict the traffic flow in the future accurately and timely as well as help the traffic management department to monitor the traffic condition in real time. It is an effective way to relieve urban traffic congestion and improve traffic efficiency. The core of traffic flow prediction is traffic data, such as vehicle speed, lane occupancy rate, and traffic flow. The transportation department can deploy the IoT devices (e.g., roadside sensor systems, embedded sensors, vehicles, or auxiliary road infrastructure) in a distributed way to collect the traffic data [2]. Then, based on the results of these data analysis, active traffic control, and intelligent route planning are carried out to avoid traffic congestion and improve traffic conditions.

Traffic flow prediction is mainly based on historical traffic data and the topology of the traffic network to predict future traffic conditions, such as traffic flow and congestion. Generally, traffic flow prediction mainly adopts the cloud computing. The IoT devices collect the data and upload them to the cloud server for training the model (e.g., [3] and [4]). Accurate and real-time traffic flow prediction faces many challenges due to complex temporal and spatial dependencies as well as inherent long-term prediction difficulties in hybrid urban transportation network topological structures. In traffic flow prediction, traffic data is based on the non-Euclidean space

2327-4662 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

and there are dynamic dependencies among the traffic flow of neighbor nodes. Usually, traditional deep learning methods [e.g., convolutional neural networks (CNNs)] cannot model the feature correlations, simultaneously. Therefore, in order to improve the accuracy of traffic flow prediction, researchers have proposed a graph convolutional neural network (GCN) to represent the spatial correlation of graph data, which improved the prediction performance [5]. As a traditional deep learning method, CNN has great advantages in extracting features of the Euclidean space data. But for the non-Euclidean space data, such as graph structure data, the prediction effect of traditional deep learning methods (e.g., CNNs) is not ideal. Since the graph is irregular, each graph may have variable size of disorder nodes. Moreover, each node in the graph may have a different number of neighbor nodes. Therefore, some important operations (e.g., convolution operations) are easy to calculate on the image, but it is no longer suitable for direct use in the graph [6]. The graph neural network (GNN) can the handle non-Euclidean space data (e.g., graph structure data) well, and each data sample (node) has an edge related to other data samples (nodes) in the graph, which is used to capture the interdependence among the data samples. At present, many methods that use GCN to predict traffic flow have achieved good performances, but these methods only cover the 1-order neighbor nodes and do not consider the influence of indirect neighbor nodes on current nodes. Meanwhile, they cannot cover indirect neighbor nodes and have weaker expression ability.

In order to deal with aforementioned challenges, in this article, we propose a novel spatial-temporal Chebyshev GNN model (ST-ChebNet) for traffic flow prediction. ST-ChebNet combines the Chebyshev GNN and long short-term memory (LSTM) model, which can not only capture the spatial dependencies and temporal dependencies of traffic data, but also consider the influence of indirect neighbor nodes on current nodes. Moreover, ST-ChebNet optimizes the relationship between nodes to realize accurate traffic flow prediction.

The main contributions can be summarized as follows.

1) We integrate the Chebyshev GNN (ChebNet) model into the traffic network, and the K-order convolution operator of which can cover the K-order neighbor nodes of the current node. Moreover, the influence of indirect neighbor nodes is considered for traffic flow to capture the spatial features of the traffic network, comprehensively.
2) We add a fully connected layer to fuse the features of traffic data, and then the LSTM model is used to extract the temporal features. Since the output of the LSTM model depends on the input of the current neurons and it will be affected by the state of the previous cell, LSTM can better extract temporal features.
3) Extensive experiments are conducted on real-world highway traffic data sets, which show that ST-ChebNet is better than the existing baseline methods in realizing accurate traffic flow prediction.

The remainder of the article is organized as follows. Section II introduces the related work. Section III presents the preliminaries of our proposed model ST-ChebNet. Then, Section IV proposes spatial-temporal Chebyshev GNN in detail. In Section V, we conduct a series of experiments and evaluate the performances of ST-ChebNet. Finally, Section VI summarizes this article.

II. Related Work

In this section, we briefly introduce the related work in traffic flow prediction. Generally, the traffic flow prediction methods mainly include three categories: 1) traditional statistical methods; 2) traditional machine learning methods; and 3) deep learning methods.

Traditional Statistical Methods: Traditional statistical methods are mainly based on mathematical statistics to make traffic flow prediction. They simply deploy time series models, which include the auto-regressive integrated moving average (ARIMA) [7], Kalman filtering model [8], etc. The stationarity of the time series is the starting point of ARIMA. Kalman filtering model adopted a state space that was defined through an observation and a state equation to filter the noise so that it could make accurate traffic flow prediction. Sang and Li [9] utilized the autoregressive moving average model (ARMA) and Markov modulated Poisson process (MMPP) to explore the deterioration of the prediction accuracy in long-range predictions. However, these models only apply to relatively stable and linear traffic flow prediction and cannot meet the actual application requirements.

Traditional Machine Learning Methods: Different from traditional statistical methods, traditional machine learning methods can model dynamic and nonlinear traffic data, and then make traffic flow prediction, but the prediction is not ideal. In [10], the support vector regression (SVR) was presented to deal with the short-term traffic forecasting issues. In addition, K-nearest neighbors (K-NN) [11] and the Bayesian method [12] were also used to model the traffic data and predict the traffic flow. However, these algorithms can model more complex features of traffic flow, but their ability to capture nonlinear patterns is limited and they are not suitable for processing irregular graph data. Therefore, traditional machine learning methods cannot handle complex and highly nonlinear data and they cannot mine deep and implicit temporal and spatial correlations from large-scale traffic data. It is difficult to improve the accuracy of the model effectively.

Deep Learning Methods: In order to solve the nonlinear and complex patterns problems, deep learning methods are widely used in traffic flow prediction. Lv *et al.* [3] proposed a deep learning approach that used the stacked autoencoder (SAE) model to capture the traffic data features, which could achieve accurate prediction. Koesdwiady *et al.* [13] improved the traffic flow prediction based on the weather information in connected cars. However, the application range of these algorithms is relatively limited. In [14], the gated recurrent units (GRUs), LSTM, recurrent neural network (RNN), and CNN were integrated to learn the spectral-spatial-temporal features in multispectral imagery. But these model are limited, which can only be applied to the Euclidean structures. Peng *et al.* [15] proposed a novel spatial-temporal incidence dynamic GNNs to capture the spatio-temporal features of traffic flow, which could make the prediction accurately.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

Ma *et al.* [16] deployed CNN to extract the patterns of traffic flow, and then they fed the extracted features into LSTM units to learn the intraday temporal evolution of traffic flow, which enhanced the prediction performance. In [17], a hybrid LSTM was presented, which divided the short-term traffic flow prediction into two stages to obtain the accurate prediction. Guo *et al.* [18] presented a deep spatial-temporal 3-D CNNs for traffic raster data prediction, which used 3D-CNN to capture the spatial-temporal correlations automatically, and then they aggregated them to obtain the final prediction. Chen *et al.* [19] adopted the graph to represent the traffic data based on spatial-temporal GNNs (STGNNs) [6]. Zhang *et al.* [20] proposed a novel framework, structure learning convolution (SLC), which was an extension of the traditional CNN. Meanwhile, SLC learned the graph structure to realize traffic flow prediction. In recent years, the graph-driven traffic flow prediction (e.g., traffic flow prediction methods based on GNNs) has developed rapidly, which can handle the non-Euclidean space data well and make traffic flow prediction more accurately. The complex traffic network is a typical non-Euclidean structure. Therefore, GNNs show good performances in traffic flow prediction.

GCNs are spectrum-based GNNs, which can extract the spatial features of the traffic topology network and convolve the graph-structured data effectively [21]. Therefore, GCNs are widely used in spatial-temporal traffic flow prediction [22]. Then, researchers have proposed many methods based on GCNs. GCNs were originally proposed by Bruna, which connected the deep neural networks and spectral graph theory [23]. Subsequently, Li *et al.* [24] proposed a diffusion convolutional RNN (DCRNN), which used graph convolutional network to predict traffic flow and reformulated the spatial dependence of traffic as a diffusion process. DCRNN is mainly aimed at long-term traffic speed prediction. Subsequently, Yu *et al.* [25] proposed a spatial-temporal graph convolutional networks (STGCNs) model, which was based on CNN to solve the time series prediction problem in the transportation field. Then, Bai *et al.* [26] proposed a sequence-to-sequence model for multistep passenger demand forecasting (STG2Seq), which used a gated graph convolution module to combine two attention mechanisms to capture temporal and spatial relationships. Zhao *et al.* [27] proposed a neural network-based traffic forecasting method (T-GCN), which combined the GCN with GRU to capture both spatial and temporal dependences simultaneously. Then, Zhu *et al.* [28] introduced an attention mechanism that could capture the global temporal dynamics and spatial correlations, enhancing the performance of T-GCN. Guo *et al.* [5] proposed an attention mechanism-based spatial-temporal GCNs for traffic flow forecasting (ASTGCN). Guo *et al.* [29] added a self-attention mechanism to the STGNN, which captured the dynamics of traffic data and provided more accurate long-term prediction. Moreover, Wu *et al.* proposed a novel GNN architecture, Graph WaveNet, for spatial-temporal graph modeling that could better process long-range time series. In addition, an adaptive adjacency matrix was designed to capture hidden spatial correlation [30]. However, the GCNs did not take the indirect neighbor nodes into account and the expression ability of them was weak.

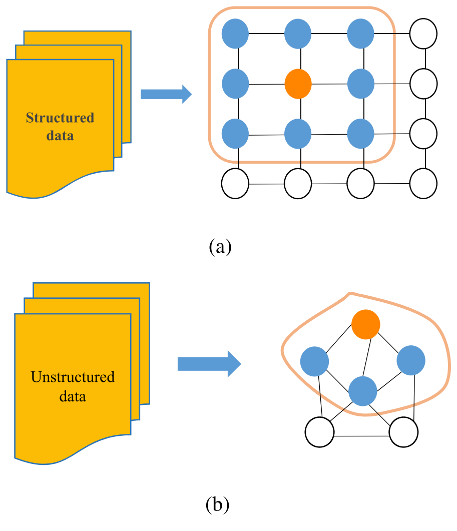


Fig. 1. Different data structures (a) Grid-like data. (b) Traffic data.

III. Preliminaries

Different from general structured data with a regular data structure (e.g., images) as shown in Fig. 1(a), traffic data is unstructured and can be transformed into graphs as shown in Fig. 1(b). Generally, a graph denotes a collection of nodes and edges, which represents a complex topological structure with spatial features. In the traffic network, the collection of intersections and roads can construct a complex traffic network, where intersections denote nodes in the graph and roads denote the edges in the graph. Then, the traffic network can be transformed into a graph structure. The structured data can use traditional CNN to process, which has a better convolution effect on the structured data structure. However, for the unstructured data structure as shown in Fig. 1(b), CNNs have many drawbacks, thus the researchers and scholars proposed many models for processing the unstructured data structure, meanwhile, the spatial convolution models were generated.

Generally, there are one-way roads in the traffic network that may affect the traffic flow prediction, but the number of which is very small in the city and can be ignored. Even if it is a one-way road, constructing an undirected graph may has a different impact on the adjacent roads of the current road, but in a complex traffic network, the different effect has less impact on the traffic flow prediction. Moreover, most roads are bidirectional such as crossroads. One road generally can affect the other three roads. In the traffic network, nodes influence each other and the influence of the direction between the two roads can be offset. Therefore, we construct the traffic network as an undirected graph.

Traffic Network: In ST-ChebNet, we define the traffic network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, 2, \ldots, i\}(1 \leq i \leq \mathcal{N})$ is a finite set of $\mathcal{N}$ nodes where each node i represents a traffic monitoring facility (e.g., traffic observation station or traffic detector); $\mathcal{E} = \{e_{ij}\}$ is a set of edges, which represent the connection between nodes i and j; and $\mathbf{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ represents the adjacency matrix of graph $\mathcal{G}$.

Traffic Flow Prediction: Assume that each node in the traffic network $\mathcal{G}$ records the fth time series that are the traffic flow

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

sequence, where $f \in \{1, 2, \ldots, \mathcal{F}\}$ and $\mathcal{F}$ denotes the number of time series. In the same sampling frequency, there are $\mathcal{F}$ measurements at each node on the traffic network $\mathcal{G}$, that is, at each time slice, the feature vector with length $\mathcal{F}$ will be generated by each node. Moreover, the symbol $x_t^{j,c} \in \mathbb{R}$ means the value of the cth feature of node j at time t. $\mathbf{x}_t^j \in \mathbb{R}^{\mathcal{F}}$ indicates all the features of node j at time t. And all the features of all nodes at time t are denoted as $\mathbf{X}_t = (\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^{\mathcal{N}})^T \in \mathbb{R}^{\mathcal{N}\times\mathcal{F}}$. And $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\vartheta)^T \in \mathbb{R}^{\mathcal{N}\times\mathcal{F}\times\vartheta}$ represents the features of all the nodes over ϑ time slices. Then, let $y_t = x_t^{j,f} \in \mathbb{R}$, which is the traffic flow at time t on jth node in the future.

Problem: Given $\mathcal{X}$, all the historical measurements of all the nodes over past ϑ time slices, we can predict the sequences of traffic flow of all nodes on the whole traffic network over next $\mathcal{T}_p$ time slices in the future, which are denoted as $\mathbf{Y} = (\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{\mathcal{N}})^T \in \mathbb{R}^{N\times\mathcal{T}_p}$.

IV. Spatial-Temporal Chebyshev Graph Neural Network

Traffic flow prediction is based on the time series of historical traffic flow data to predict the traffic flow in the future, that is, at time t, the traffic flow of next time $t+\Delta t$ can be predicted, where Δt denotes time interval between two measurements. Generally, traffic flow prediction methods can be divided into two categories according to the length of prediction time: 1) short-term traffic flow prediction and 2) medium and long term traffic flow prediction [25]. The prediction time of short-term traffic flow prediction does not exceed 15 min, which is highly nonlinear and uncertain. It has a strong correlation in terms of time and it is suitable for scenarios with high real-time requirements. For the medium and long-term traffic flow prediction, the prediction time is more than 30 min. The key for traffic flow prediction is to ensure real time. In a real-traffic network, if the time period predicted by the traffic flow prediction model is too long, the vehicles may choose the same smoother route when traffic congestion occurs, and all vehicles have enough time to drive to the smoother route. If the vehicles choose the smoother route, then the smoother route will also cause traffic congestion. Therefore, the prediction time for traffic flow should not be too long. In traffic flow prediction, real time is a key indicator that reflects traffic conditions, so our prediction time is set no more than 5 min, which is convenient for the driver to make the best route choice. Thus, in this article, we mainly focus on short-term traffic flow prediction. We will predict the traffic flow when Δt is no more than 5 min. Meanwhile, the changes of traffic flow are extremely complex and exhibit unique characteristics, such as uncertainty, nonlinearity, randomness, correlation, and periodicity, which show that the short-term traffic flow system is changeable and complex, bringing great difficulty to the traffic flow prediction. However, its own unique regularity of short-term traffic flow system provides the possibility for traffic flow prediction. Here, we propose a short-term traffic flow prediction model based on graph-driven machine learning, which is of the great significance.

In the traffic network, the vehicles can know the traffic flow information of roads in real time at a certain time Δt in the future through traffic flow prediction, which can avoid traffic congestion for vehicles and facilitate travel for users. Next, we will introduce ST-ChebNet in detail.

A. Overview

In this section, we introduce ST-ChebNet for traffic flow prediction. In short-term traffic flow prediction, the extraction of spatial features and temporal features between intersections in the traffic network is very challenging. Concretely, the ST-ChebNet is mainly composed of the LSTM model and ChebNet model. As shown in Fig. 2, we first collect the traffic data of roads and intersections from the complex traffic network to form an undirected graph, which takes the roads as edges and the intersections as nodes. Then, the traffic monitoring facilities (e.g., traffic detectors) collects the information of traffic flow, lane occupancy rate, and vehicle speed to predict the traffic flow in the future. Generally, in ST-ChebNet, the traffic monitoring facilities collect data every Δt minutes and perform multisegment measurements to obtain more historical data. Then, the obtained data is fed into the ST-ChebNet for training. First, we use a fully connected layer to fuse the features. Then, we use the LSTM model [31] to capture the temporal features. Next, the ChebNet model [32] further captures the spatial features. Through the activation function, we can obtain the prediction results that denote the traffic flow of the traffic network in the future. In Fig. 2, $\hat{Y}_{t+\Delta t}$ and $Y_{t+\Delta t}$ denote the predict value and real value at $t+\Delta t$, respectively.

B. Traffic Data Collection

With the rapid development of the computer technology, the collection methods of short-term traffic flow are becoming more and more perfect and abundant. The previous collection method generally relied on the artificial observation, but the current collection methods are more automated and intelligent. According to the location where the traffic detectors are placed, the collection methods of short-term traffic flow mainly include two methods: 1) the mobile short-term traffic flow collection method and 2) the fixed short-term traffic flow collection method.

1) *Mobile Short-Term Traffic Flow Collection Method:* The mobile short-term traffic flow collection method uses mobile vehicles equipped with sensors or detectors to monitor fixed markers on the road to obtain the traffic data. Presently, the following four methods are widely used: a) the dynamic road traffic data collection method based on the vehicles with GPS; b) the dynamic road traffic data collection method based on RFID; c) the dynamic road traffic data collection method based on automatic identification of vehicles licenses; and d) the traffic information collection method based on probe vehicles.
2) *Fixed Short-Term Traffic Flow Collection Method:* The fixed short-term traffic flow collection method uses a traffic detector installed in a fixed location to monitor the moving vehicles, so as to realize the collection of

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

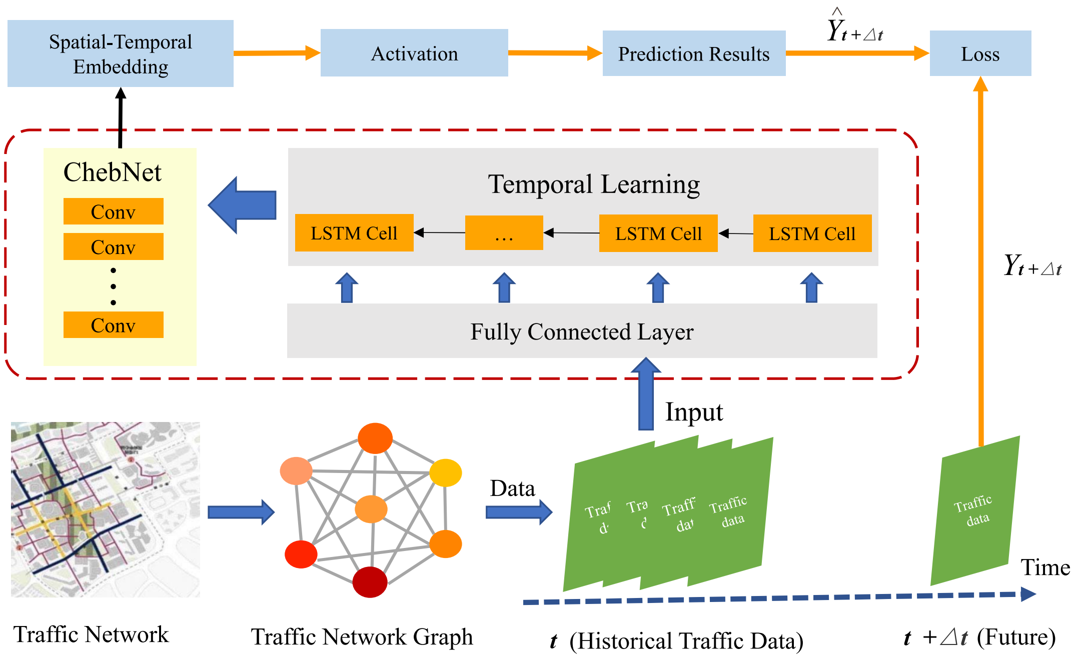


Fig. 2. Architecture of ST-ChebNet.

traffic data. Meanwhile, fixed short-term traffic flow collection method can monitor the traffic data of fixed node. Usually, the traffic detectors are installed at intersections or roads.

Using the above two traffic flow collection methods, the collected traffic data can be used to denote the features of intersections or roads during a certain period of time.

The lane occupancy rate, traffic flow, and vehicle speed are important metrics for measuring traffic conditions. The mutual influence of metrics has great guiding significance for improving traffic conditions, preventing traffic congestion, improving transportation efficiency, and management as well as realizing reasonable distribution of traffic flow. Specifically, there are two main factors affecting traffic flow.

1) *Land Utilization Rate:* The use of urban land will affect the occurrence and attraction of traffic. In densely populated places, the occurrence and attraction of traffic will be very large.
2) *Weather Conditions:* Weather conditions also have a great impact on traffic flow. Rain or a bad weather will affect the occurrence and attraction of traffic in a certain area, which will be directly reflected by the lane occupancy rate, vehicle speed, and traffic flow.

The lane occupancy rate refers to the proportion of time that the vehicles are in the detector control area. The vehicle speed indicates the speed of the vehicle. Traffic flow represents the number of vehicles passing through the lane in a unit time. When the lane occupancy rate is less than the optimal lane occupancy rate, the traffic flow is in free driving and the average vehicle speed is higher. When the lane occupancy rate is close to or equal to the optimal lane occupancy rate, the traffic flow will follow the vehicle and the speed will decrease. At this time, all kinds of vehicles keep driving at a constant speed and the traffic flow will reach the maximum. When the lane

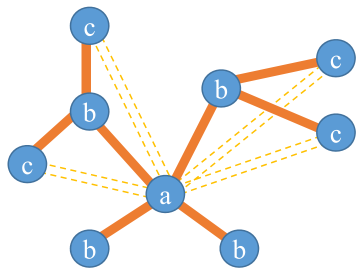


Fig. 3. Nodes distribution.

occupancy rate is greater than the optimal lane occupancy rate, the traffic flow is in a congested state. Then, the traffic flow gradually increases, the vehicle speed and traffic flow decrease at the same time. Finally, the traffic congestion occurs.

In this article, we adopt the fixed short-term traffic flow collection method to collect the traffic data. Suppose that the intersections are regarded as nodes, the features of traffic flow, vehicle speed, lane occupancy rate on each node in the traffic network can be obtained. Meanwhile, the distance of intersections denotes the weight between two nodes and we construct an undirected graph for the traffic network.

C. Nodes Correlation Analysis in Traffic Network

In the traffic network, the traffic flow in the indirect neighbor nodes of a node has an impact on the traffic flow in the node. Therefore, in the traffic flow prediction, the indirect neighbor nodes also play an important role. Fig. 3 is the nodes distribution in traffic network. Taking intersection a as the central node, nodes b are 1-order neighbor nodes of intersection a, and nodes c are the 2-order neighbor nodes of intersection a. The solid lines represent the road. The traffic flow of all nodes b will have a direct impact on the traffic flow at node a. And the traffic flow at nodes c will affect the traffic flow at nodes b.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

In the urban traffic network, the traffic flow of nodes b affects the traffic flow of node a, and the traffic flow of nodes c will also affect the traffic flow of node a indirectly. In Fig. 3, the indirect impacts are represented by dotted lines. In order to verify our inference, we give the following analysis. In order to simplify the calculation, we prove that the 2-order neighbor nodes can affect the current node. Then, we take the PeMS04 data set as an example, which is introduced in Section V-A in detail.

We take out the 2-order nodes that are adjacent to the node 1 in the PeMS04 data set for correlation analysis. From the PeMS04 data set, we can obtain the 2-order nodes of node 1, that are node 252, node 119, node 27, node 216, node 31, node 250, node 70, node 67, node 217, node 122, and node 120.

Then, we calculate the covariance S_{xy} between the node 1 and its 2-order neighbor nodes by (1), where $\mathbb{X}_i$ denotes the change of traffic flow between current measurement and previous measurement at the central node and $\bar{\mathbb{X}}$ means the average value of all $\mathbb{X}_i$s at the central node. $\mathbb{Y}_i$ represents the sum of the changes in traffic flow between adjacent measurement at all 2-order neighbor nodes and $\bar{\mathbb{Y}}$ denotes the average value of changes in traffic flow among all measurements at all 2-order neighbor nodes. n_t denotes the times of changes in traffic flow. Moreover, x represents the central node and y represents the set of all 2-order neighbor nodes

$$S_{xy} = \frac{\sum_{i=1}^{n_t} (\mathbb{X}_i - \bar{\mathbb{X}})(\mathbb{Y}_i - \bar{\mathbb{Y}})}{n_t - 1}. \quad (1)$$

The standard deviation S_x of traffic flow changes at node 1 can be calculated through

$$S_x = \sqrt{\frac{\sum (\mathbb{X}_i - \bar{\mathbb{X}})^2}{n_t - 1}}. \quad (2)$$

Next, we calculate the standard deviation S_y of the sum of changes of traffic flow at all 2-order neighbor nodes through

$$S_y = \sqrt{\frac{\sum (\mathbb{Y}_i - \bar{\mathbb{Y}})^2}{n_t - 1}}. \quad (3)$$

Finally, through (4), we can calculate that the correlation coefficient r_{xy} of the traffic flow between node 1 and its 2-order neighbor nodes is 0.3, which is greater than 0. Therefore, node 1 and its 2-order neighbor nodes have a weak correlation. Although the weak correlation cannot play a decisive role in traffic flow prediction, it can still become a key feature in traffic flow prediction

$$r_{xy} = \frac{S_{xy}}{S_x S_y}. \quad (4)$$

D. Data Set Partition

Assuming that the current time is t_0, the sensor nodes or traffic detectors sample q times every 60 min, and the prediction window size is $\mathcal{T}_p$. The length of the time series that we intercept is $\mathcal{T}_h$, which is an integer multiple of $\mathcal{T}_p$. In ST-ChebNet, we use the traffic flow in $\mathcal{T}_h$ time slices to predict the traffic flow in $\mathcal{T}_p$ time slices. $\mathcal{X} = (\mathbf{X}_{t_0-\mathcal{T}_h+1}, \mathbf{X}_{t_0-\mathcal{T}_h+2}, \ldots, \mathbf{X}_{t_0} \in \mathbb{R}^{\mathcal{N}\times\mathcal{F}\times\mathcal{T}_h})$ is a historical time series, which is directly adjacent to the prediction period. In the real traffic, with the increase of vehicles, traffic congestion often occurs, seriously reducing travel efficiency. Generally, it will take some time or even a long time for traffic congestion to resume normal traffic. Therefore, the recent traffic flow will inevitably have a great impact on the future traffic flow. The traffic data is combined, which will be regarded as the samples for model training.

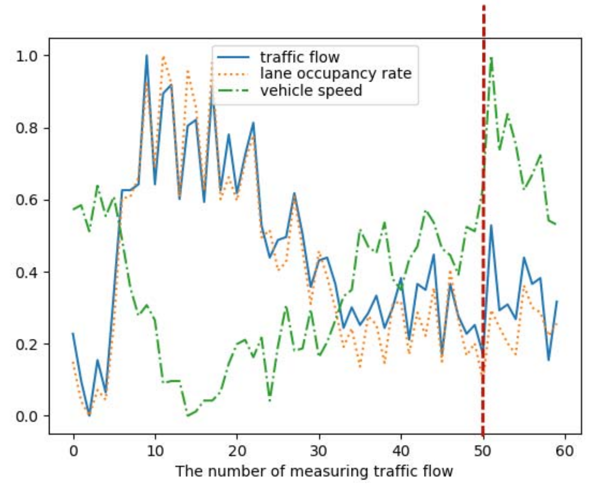


Fig. 4. Time series of traffic flow at a intersection.

E. Traffic Flow Prediction

In this section, we describe the detailed steps of traffic flow prediction, which mainly include three parts: 1) node features fusion; 2) temporal dependence modeling; and 3) spatial dependence modeling.

Node Features Fusion: We collect three features on each node that include traffic flow, lane occupancy rate, and vehicle speed, respectively. Suppose that the features of the jth node at time t are traffic flow $x_t^{j,1}$, lane occupancy rate $x_t^{j,2}$, and vehicle speed $x_t^{j,3}$. As shown in Fig. 4, it analyzes the impact of the three features on the traffic flow of a node at the future time. Fig. 4 shows that in 0 to 50 measurements, the lane occupancy rate is proportional to traffic flow, but vehicle speed is inversely proportional to traffic flow. In the future, traffic flow, lane occupancy rate, and vehicle speed measured in previous 0 to 50 measurements still have impact on traffic flow, but the degree of impact is also different.

Due to the different degrees of correlation among three features, before feeding the data into the model, the ST-ChebNet first fuses these three features on each node into one new feature, which is denoted as $x_t^{\prime j}$. Then, we assign different weights w for the three features and perform a weighted summation through the following (5). Specifically, we add a fully connected layer to assign the weight w for each feature on each node, and then the weighted features of $\mathcal{X}$ among all nodes $\mathcal{N}$ in n samples are fused into $n \times \mathcal{N} \times [(\mathcal{T}_h)/[(60/q)]]$ features, generating matrix $V = (n, \mathcal{N}, \mathcal{X}')$, where $\mathcal{X}'$ denotes features

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

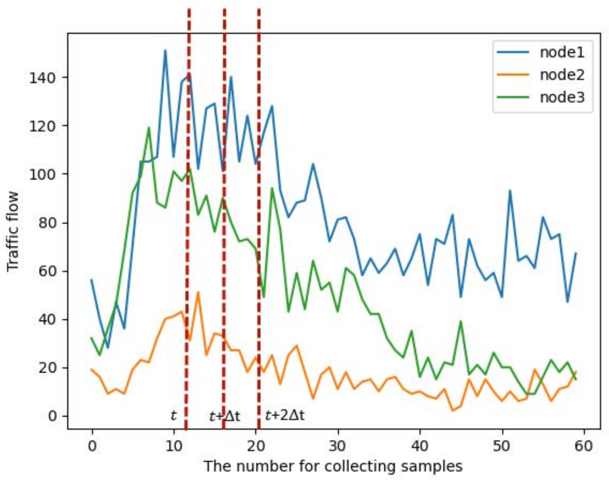


Fig. 5. Influence of the three features for the traffic flow.

of all nodes measured during $\mathcal{T}_h$ time series length

$$x_t^{\prime j} = \sum_{c=1}^{c} w_t^{j,c} x_t^{j,c} + b_t. \tag{5}$$

In (5), $x_t^{\prime j}$ denotes a new feature on the node j at time t. $w_t^{j,c}$ and $x_t^{j,c}$ denote the weight and its value of the cth feature on the node j at time t, respectively. Moreover, the b_t means the bias at time t.

Temporal Dependence Modeling: In short-term traffic flow prediction, traffic flow prediction is closely related to time series, which is the key metric to measure traffic conditions. In the natural language processing, many schemes are based on the RNN to extract the relevance in time dimension. However, the RNN usually faces many issues, such as gradient explosion and gradient disappearance. Compared with RNN, LSTM is an extension of RNN, which can effectively solve the above issues in traditional RNN. Moreover, its prediction accuracy is more accurate than RNN. Therefore, we adopt LSTM to exact the temporal features.

As shown in Fig. 5, it is the changes of three nodes over 60 times measurements for the traffic flow. The fluctuation of traffic flow between time t and $t + \Delta t$ is always within a certain range and the changes are relatively stable. But there are also times when the traffic flow changes rapidly. For this situation, it may be caused by special circumstances (e.g., traffic accidents). In brief, the traffic features $\mathbf{X}_t$ of all nodes at the time t will affect the traffic features $\mathbf{X}_{t+\Delta t}$ at the next time $t + \Delta t$. Therefore, ST-ChebNet uses the LSTM model to construct a short-term traffic flow prediction model to extract $\mathcal{T}_h$ time series and uses the ChebNet model to perform spatial correlation analysis.

In ST-ChebNet, each sample includes all node features of traffic network with $\mathcal{T}_h$ time series and the features of node at time t has a strong correlation in terms of time series with node features at $t + \Delta t$, thus, we adopt the LSTM model to extract the time-series features of the traffic network.

The LSTM model established here includes four layers, which are one input layer, two hidden layers, and one output layer, as shown in Fig. 6. And the two hidden layers are connected with each other, which are used to capture the features of the time series. Furthermore, the obtained matrix $V = (n, \mathcal{N}, \mathcal{X}')$ is fed into the LSTM model and the output is V'. Next, we mainly introduce the hidden layers, which are the core layers in LSTM model. When the $\mathcal{T}_h$ time series are input to the hidden layer h_t, the following (6)–(10) will be executed in sequence

$$\mathfrak{f}_t = \sigma\left(x_t U^{\mathfrak{f}} + h_{t-1} W^{\mathfrak{f}} + b_f\right). \tag{6}$$

In (6), $\mathfrak{f}$ denotes the forget gate, $W^{\mathfrak{f}}$ is the weight of the previous hidden layer h_{t-1}, and $U^{\mathfrak{f}}$ is the weight of current traffic flow at time t. x_t denotes the traffic flow at time t. The symbol $\mathfrak{f}_t$ controls that how much traffic flow of the cell state C_{t-1} at the previous moment will flow into the cell state C_t

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i + b_i\right) \tag{7}$$
$$\hat{C}_t = \tanh\left(x_t U^g + h_{t-1} W^g + b_C\right) \tag{8}$$
$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t. \tag{9}$$

The traffic flow at time t is input by (7), where i represents the input gate, W^i is the weight of the hidden layer h_{t-1}, U^i is the weight of the traffic flow, σ is a sigmoid function, i_t controls that how much current traffic flow is integrated into the cell state C_t. In (8), the tanh layer creates a candidate value for the new state $\hat{C}_t$. Then, we can calculate $f_t * C_{t-1} + i_t * \hat{C}_t$. Then, C_{t-1} is updated to the new cell state C_t by (9)

$$o_t = \sigma\left(x_t U^o + h_{t-1} W^o + b_o\right) \tag{10}$$
$$h_t = o_t * \tanh(C_t). \tag{11}$$

Equation (10) is the output gate, which determines the traffic flow information to be output. First, it passes through the sigmoid function, which produces an output between 0 and 1. Then, the unit state is put into tanh layer that produces an output between −1 and 1, which multiplies the output of the output gate and finally outputs the time features as shown in (11). After the hidden layer, it is the activation function as shown in (12), which is used to add nonlinear factors to improve the accuracy for the LSTM model. Meanwhile, the activation function can also solve problems that cannot be solved by the linear model

$$x_t = \mathrm{ReL}\,U\left(h_t W^d + b\right). \tag{12}$$

Next, we feed the matrix V into the LSTM model, which will output the matrix V'. And then, we feed the obtained matrix V' into the ChebNet model to further extract the spatial features for the traffic flow prediction.

Spatial Dependence Modeling: If decomposing the traffic network, and then using the traditional neural network model (e.g., the CNN model) to extract features of the traffic network, it will destroy the natural topology of the traffic network and lose the significance of the topology. However, transforming the traffic network into a topological graph can make full use of the features of the topological graph to predict the traffic flow, which exerts the potential value of the topological graph. In recent years, the GNN can better model the correlation of graph-based traffic networks. Also, it can combine

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

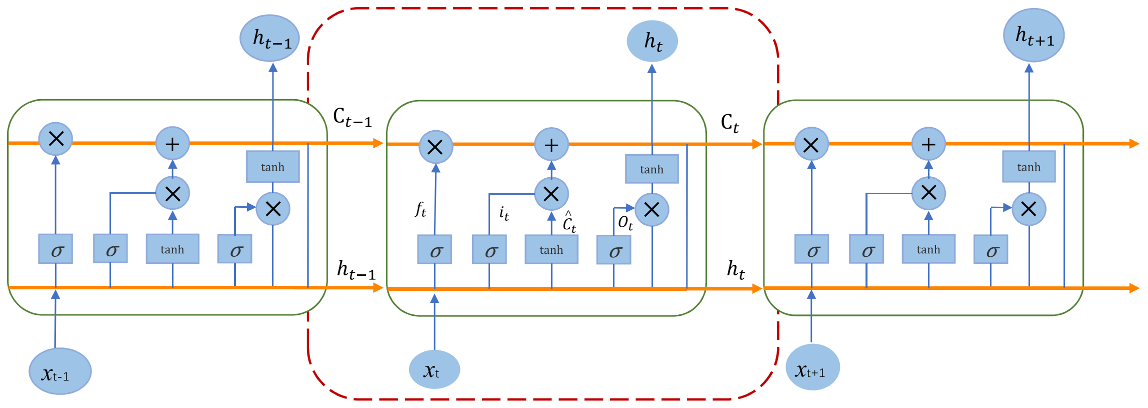


Fig. 6. Architecture of the LSTM model.

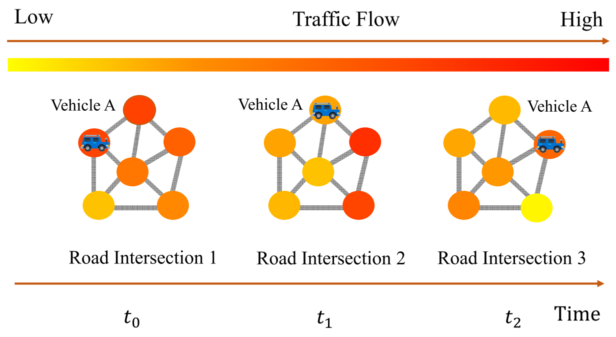


Fig. 7. Spatial distribution of the traffic network.

different types of neural networks to predict traffic flow collaboratively. Generally, the ChebNet model solves the problem that the graph convolution kernel is global and has a large number of parameters as well as reduces the complexity of graph convolution operations. Moreover, the expression of the ChebNet model is strong and the K-order convolution operator of the ChebNet model can cover the node's K-order neighbor nodes. Meanwhile, the ChebNet model does not need to perform feature decomposition for the Laplacian matrix, which greatly saves time. Therefore, we choose the ChebNet model to extract the spatial features of the traffic data.

As shown in Fig. 7, it is the spatial distribution of the traffic network. As time goes on, the traffic flow at each intersection is different. If a vehicle A passes the intersection 1 at time t_0, and the vehicle will appear at another intersection 2 after a period of time. Obviously, a vehicle not only affects the traffic flow at intersection 1, but also affects the traffic flow at intersection 2. Further, it shows that there is a spatial correlation between the two intersections. In a traffic network within a time period, the traffic flow is composed of all vehicles in the traffic network, and each vehicle will pass through different intersections at different times, so the spatial correlation between each intersection is very strong. The length of roads and the connection among roads are features of the traffic network. Therefore, using the spatial features of the traffic network can better predict the traffic flow.

For the spectral graph analysis, the graph Laplacian [33] is essential operator for the undirected and connected graph, which is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, and the normalized graph Laplacian is $\mathbf{L} = \mathbf{I}_{\mathcal{N}} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{I}_{\mathcal{N}}$ is the identity matrix, $\mathbf{A}$ is the adjacent matrix, and $\mathbf{D} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the diagonal degree matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. The Laplacian matrix is diagonalized by the Fourier basis $\mathbf{U} = [u_0, \dots, u_{\mathcal{N}-1}] \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, where $\{u_l\}_{l=0}^{\mathcal{N}-1} \in \mathbb{R}^{\mathcal{N}}$ is the orthogonal eigenvectors of $\mathbf{L}$, that is the graph Fourier modes. $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T$ is the eigenvalue decomposition for the Laplacian matrix, where $\Lambda = \mathrm{diag}([\lambda_0, \dots, \lambda_{\mathcal{N}-1}]) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ and $\{\lambda_l\}_{l=0}^{\mathcal{N}-1} \in \mathbb{R}^{\mathcal{N}}$ is the ordered real non-negative eigenvalues associated with $\mathbf{L}$, known as the frequencies of the graph. Then, we define the signal all over the graph of the traffic flow at time t as $x = \mathbf{x}_t^f \in \mathbb{R}^{\mathcal{N}}$. And the graph Fourier transform of the signal $x \in \mathbb{R}^{\mathcal{N}}$ is $\hat{x} = \mathbf{U}^T x \in \mathbb{R}^{\mathcal{N}}$, and its inverse Fourier transform as $x = \mathbf{U}\hat{x}$ [34]. Generally, in the Fourier domain, the convolution operator on graph $\mathcal{G}$ is denoted as $x_{*\mathcal{G}}y = U((U^T x) \odot (U^T y))$, where $\odot$ means the element-wise Hadamard product. The signal x on the graph $\mathcal{G}$ can be filtered by g_θ

$$y = g_\theta(\mathbf{L})x = g_\theta\big(\mathbf{U}\Lambda\mathbf{U}^T\big)x = \mathbf{U}g_\theta(\Lambda)\mathbf{U}^T x. \tag{13}$$

However, in the ChebNet model, it mainly uses Chebyshev polynomials to replace the convolution kernel in the spectral domain to represent the filter, as defined in (14), which is the truncated expansion of order $K-1$

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\Lambda}). \tag{14}$$

In (14), the parameter $\theta \in \mathbb{R}^K$ is the vector of Chebyshev coefficients. $T_k(\tilde{\mathbf{\Lambda}}) \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the Chebyshev polynomial with order k evaluated at $\tilde{\mathbf{\Lambda}} = 2\Lambda/\lambda_{\max} - \mathbf{I}_{\mathcal{N}}$, where $\tilde{\mathbf{\Lambda}}_{ii} \in [-1, 1]$ and $\lambda_{\max}$ denotes the maximum eigenvalue of the Laplacian matrix.

Moreover, the calculation of the Chebyshev polynomial $T_k(x)$ can be obtained through the following recursive equation:

$$\begin{cases} T_0(x) = 1 \\ T_1(x) = x \\ T_{\mathcal{N}+1}(x) = 2xT_{\mathcal{N}}(x) - T_{\mathcal{N}-1}(x). \end{cases} \tag{15}$$

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

Therefore, the convolution operation is expressed as

$$x_{*\mathcal{G}}g_\theta = \mathbf{U}\left(\sum_{i=0}^{K-1}\theta_k T_k(\tilde{\boldsymbol{\Lambda}})\right)\mathbf{U}^T x. \tag{16}$$

Then, similar to $\tilde{\boldsymbol{\Lambda}} = 2\boldsymbol{\Lambda}/\lambda_{\max} - \mathbf{I}_{\mathcal{N}}$ and $T_k(\tilde{\boldsymbol{\Lambda}})$, $\tilde{\mathbf{L}}$ and $T_i(\tilde{\mathbf{L}})$ can be defined in the following (17) and (18), respectively:

$$\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I}_{\mathcal{N}} \tag{17}$$

$$\begin{aligned} T_k(\tilde{\mathbf{L}}) &= T_k(2\mathbf{L}/\lambda_{\max} - \mathbf{I}_{\mathcal{N}}) \\ &= T_k\left(2\mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T/\lambda_{\max} - \mathbf{U}\mathbf{I}_{\mathcal{N}}\mathbf{U}^T\right) \\ &= T_k\left(\mathbf{U}(2\boldsymbol{\Lambda}/\lambda_{\max} - \mathbf{I}_{\mathcal{N}})\mathbf{U}^T\right) \\ &= \mathbf{U}T_k\left(\tilde{\boldsymbol{\Lambda}}\right)\mathbf{U}^T. \end{aligned} \tag{18}$$

Meanwhile, performing the Fourier transform, convolution and inverse Fourier transform can avoid calculating the $\mathbf{U}$ of the Fourier transform in the graph, and the convolution equation can be further simplified in

$$\begin{aligned} x_{*\mathcal{G}}g_\theta &= \mathbf{U}\left(\sum_{i=0}^{K-1}\theta_k T_k(\tilde{\boldsymbol{\Lambda}})\right)\mathbf{U}^T x \\ &= \sum_{i=0}^{K-1}\theta_k\left(\mathbf{U}T_k(\tilde{\boldsymbol{\Lambda}})\mathbf{U}^T\right)x \\ &= \sum_{i=0}^{K-1}\theta_k T_k\left(\tilde{\mathbf{L}}\right)x. \end{aligned} \tag{19}$$

From (19), we can see that the calculation process does not require eigenvector decomposition, which greatly reduces the computational complexity.

Then, given a sample s_m with jth output feature map, which is denoted as

$$y_{s_m,j} = \sum_{i=1}^{F_{\text{in}}} g_{\theta_{i,j}}(\mathbf{L})x_{s_m,i} \in \mathbb{R}^{\mathcal{N}}. \tag{20}$$

In (20), $x_{s_m,i}$ denotes the feature maps of the input and the $F_{\text{in}} \times F_{\text{out}}$ vectors in Chebyshev coefficients $\theta_{i,j} \in \mathbb{R}^K$ represent the trainable parameters of the layer. Next, the two gradients in (21) and (22) are needed to train multiple convolutional layers with the backpropagation algorithm

$$\frac{\partial B}{\partial \theta_{i,j}} = \sum_{s_m=1}^{n}\left[\bar{x}_{s_m,i,0}, \ldots, \bar{x}_{s_m,i,K-1}\right]^T \frac{\partial B}{\partial y_{s_m,j}} \tag{21}$$

$$\frac{\partial B}{\partial x_{s_m,i}} = \sum_{j=1}^{F_{\text{out}}} g_{\theta_{i,j}}(L)\frac{\partial B}{\partial y_{s_m,j}}. \tag{22}$$

In (21) and (22), B means the loss energy over a mini batch of n samples.

Then, the ChebNet model uses the coarsening phase of the Graclus multilevel clustering algorithm [35] to cluster large-scale various graphs and chooses the normalized cut [36] to compute the successive coarser versions for a graph, which can also minimize several popular spectral clustering objectives. Finally, graph coarsening will roughly divide the number of nodes by 2, from one level to the next coarsening level.

After the graph coarsening, the pooling on graph signals will be performed and we will obtain the results of spatial-temporal features fusion.

In ST-ChebNet, it mainly includes a fully connected layer, LSTM model, and ChebNet model. ST-ChebNet first introduces a fully connected layer to fuse the three features into a new feature, which can denote the information of a node. Then, the LSTM model is used to capture the temporal feature of traffic data, which mainly includes one input layer, two hidden layers, and one output layer. The matrix V' obtained in the LSTM and adjacency matrix $\mathbf{A}$ are fed into the ChebNet model for graph convolution, which can capture spatial features of traffic data and the final results are traffic flow $\hat{y}$ in the future.

F. Loss Function

The purpose of the training is to minimize the errors between the real and predicted traffic flow. In ST-ChebNet, we adopt the y to represent real traffic flow and use $\hat{y}$ to denote the predicted traffic flow. Meanwhile, the mean square error (MSE) is exploited to evaluate our model, the smaller the value of MSE, the better the accuracy of the ST-ChebNet. Then, we can calculate the MSE through

$$\text{MSE} = \frac{1}{n_{\text{test}}}\sum_{i=1}^{n_{\text{test}}}\left(y_i - \hat{y}_i\right)^2. \tag{23}$$

In the above (23), n_{test} denotes the number of test samples, y_i and $\hat{y}_i$ are the real value and the prediction value, respectively.

V. Experiments

In order to measure the prediction performance of ST-ChebNet, extensive experiments are taken on two real-world data sets: 1) PeMS04 data set and 2) PeMS08 data set, which are about the highway traffic flow.

A. Data Set Description

We use the highway traffic data sets PeMS from California, which are collected by the Caltrans Performance Measurement System [37]. PeMS data sets collected the traffic data in real-time more than 39000 detectors, which were deployed on the highway in the major metropolitan areas in Calfornia and sampled every 5 min. Moreover, PeMS data sets include the geographic information of the sensor stations. We test ST-ChebNet on two PeMS data sets, which are PeMS04 data set and PeMS08 data set, respectively. PeMS04 data set is the traffic data obtained by the 307 detectors, which was collected on January 1, 2018. And PeMS04 data set collects 59 consecutive days of traffic data. PeMS08 data set is the traffic data collected from 170 detectors, which cost about 62 consecutive days. PeMS08 data set was collected on July 1, 2016.

B. Experimental Settings

The experiments are conducted on the PeMS04 data set and PeMS08 data set, which divide all data sets into training set and test set at a ration of 3:1. Moreover, we use historical traffic data from different time periods to predict traffic flow

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

in the future. In ST-ChebNet, we set $\mathcal{T}_p = 5$ min and $\mathcal{T}_h$ is varied from 30 to 150 min, where the step is 30 min.

Also, the experiments under the operating system windows 10 professional edition with Intel Core i7 CPU (6 core 3.7 GHz) and 16-GB RAM memory. ST-ChebNet is implemented in PyTorch, which is a tensor library used to optimize the GPU and CPU. In the experiments, we take 64 samples one time for training and the Adam optimizer is adopted to optimize the loss of the training set to make the loss smaller.

C. Baseline Methods

We compare the ST-Chebnet with the following models.

CNN [38]: CNN, which is a hierarchical sequential architectures based on RNNs.

GCN [39]: GCN is a very powerful neural network architecture for graph data.

GAT [40]: Graph attention network, in which the multihead self-attention mechanism is used to automatically learn and optimize the connection relationship.

LSTM [31]: It is an RNN architecture, which is widely used in deep learning.

DCRNN [24]: DCRNN is a deep learning framework, which integrates the spatial and temporal dependency of traffic network for traffic flow prediction.

ST-GCN [25]: ST-GCN is used to tackle the time series prediction problem in the traffic domain.

ASTGCN [5]: ASTGCN is an attention-based spatial-temporal GCN, which can effectively capture the dynamic spatial-temporal features for traffic flow prediction.

Graph WaveNet [30]: Graph WaveNet designs an adaptive adjacency matrix to capture hidden spatial dependency in traffic data and it can handle long-range sequences.

D. Evaluation Metrics

In the traffic flow prediction, accuracy is a very important indicator to measure the quality of the model, and timely and accurate traffic flow prediction is essential for the traffic control and guidance. Accuracy is an essential feature of traffic flow prediction and it cannot be improved through external factors. Therefore, in this article, we choose the accuracy to analyze the experimental results. Moreover, many existing papers, such as [5], [29], [41], and [42] adopted mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RMSE) to evaluate the accuracy of their models. Here, we also use the above three metrics to measure the accuracy of ST-ChebNet.

1) *MAE:* MAE is also called L_1 norm loss, which is the average of absolute differences between prediction value and real value. We can calculate the MAE by the following (24), where y_i is the real value and $\hat{y}_i$ is the prediction value:

$$\text{MAE} = \frac{\sum_{i=1}^{n_{\text{test}}} \left|y_i - \hat{y}_i\right|}{n_{\text{test}}}. \tag{24}$$

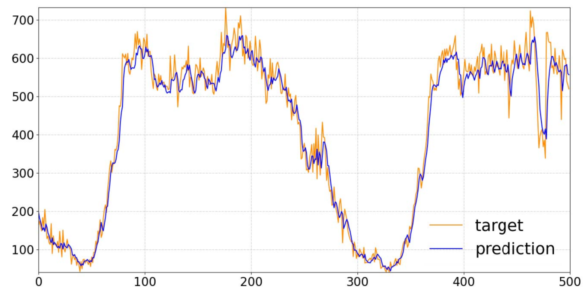


Fig. 8. Fitting effect of ST-ChebNet on PeMS04 data set.

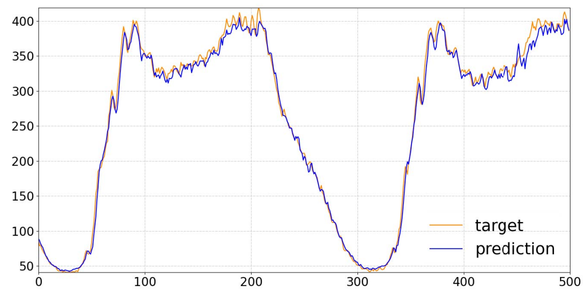


Fig. 9. Fitting effect of ST-ChebNet on PeMS08 data set.

2) *MAPE:* MAPE is an average for absolute percentage errors. If MAPE is 0%, it indicates the model is a perfect model. And if an MAPE greater than 100%, it means the model is an inferior model. MAPE is computed by the following (25), where y_i is the real value and $\hat{y}_i$ is the prediction value:

$$\text{MAPE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \frac{\left|y_i - \hat{y}_i\right|}{y_i}. \tag{25}$$

3) *RMSE:* RMSE is a measure of the differences between the predicted value and the real value, which can be calculated by (26), where y_i is the real value and $\hat{y}_i$ is the prediction value

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_{\text{test}}} \left(y_i - \hat{y}_i\right)^2}{n_{\text{test}}}}. \tag{26}$$

E. Experimental Results

In this section, we compare ST-ChebNet with the other eight baseline methods based on PeMS04 data set and PeMS08 data set.

Figs. 8 and 9 show the traffic flow of a intersection (node) on PeMS04 data set and PeMS08 data set, respectively. ST-ChebNet selects the real value of the 500 measured traffic flow of the node and predicted value for fitting. It can be concluded that ST-ChebNet has a good prediction effect on the PeMS04 data set and PeMS08 data set even if large changes in traffic flow occur. ST-ChebNet is suitable for short-term traffic flow prediction.

Fig. 10 represents the prediction performance of the ST-Chebnet model using different time series of historical data. It uses 30, 60, 90, 120, and 150 min to predict the traffic flow in

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

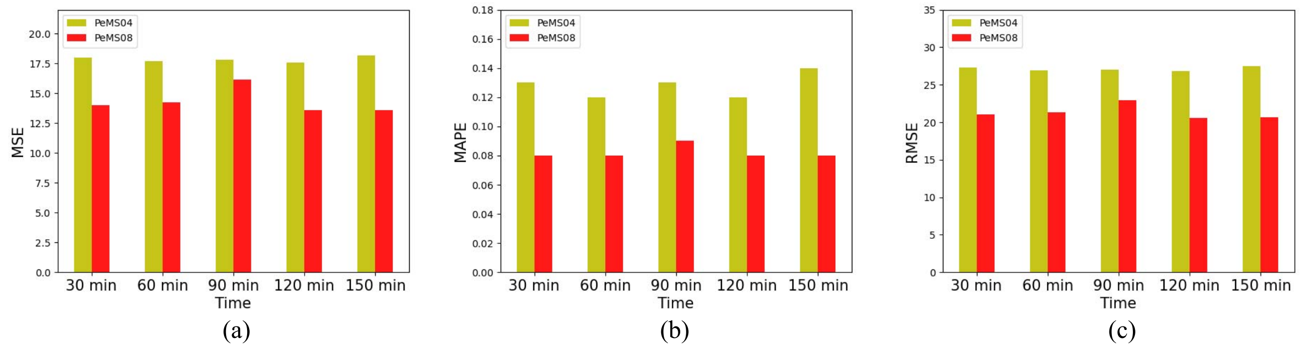


Fig. 10. Prediction results of ST-Chebnet with different time series on PeMS04 and PeMS08. (a) MAE. (b) MAPE. (c) RMSE.

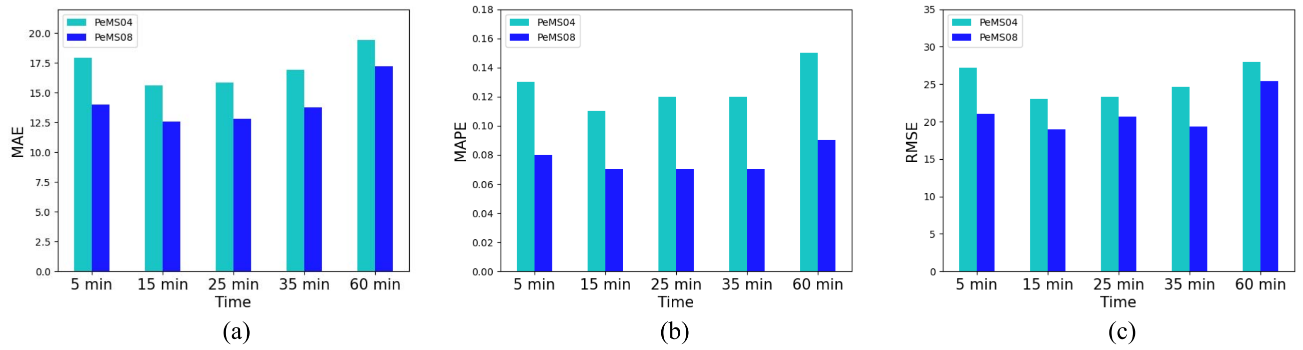


Fig. 11. Prediction results of ST-ChebNet on PeMS04 and PeMS08. (a) MAE. (b) MAPE. (c) RMSE.

the next 5 min. It can be seen from Fig. 10 that the different historical time series of the PeMS04 and PeMS08 data sets have little effect on the prediction results. In the PeMS04 data set, MAE, MAPE, and RMSE calculated from 30 to 150 min (the lower the value, the better) fluctuate little, and the same is true for PeMS08 data set. However, the MAE, MAPE, and RMSE predicted by different time series of historical data on the PeMS04 data set are higher than those of the PeMS08 data set, since the complexity of the PeMS04 data set is higher than that of the PeMS08 data set. PeMS04 has more routes and more complex traffic network, so it is more difficult. In short-term traffic flow prediction, using historical data at different times (e.g., within two and a half hours) to predict traffic flow has little effect on the prediction results. Moreover, The value of $\mathcal{T}_h$ has little effect on accuracy, and for smaller $\mathcal{T}_h$, it is easier to collect data. Therefore, ST-Chebnet sets $\mathcal{T}_h = 30$ min and collects the 30-min historical data to predict traffic flow, avoiding the cumbersome problem of collecting a large amount of historical data.

Through the analysis of Fig. 10, we choose to use 30 min of historical data to predict the traffic flow in the next 5 min. While in Fig. 11, we use 30 min to predict the traffic flow at different times, which are predicted 5, 15, 25, 35, and 60 min of traffic flow. We use the 5-min traffic flow as a unit and convert the predicted traffic flow at different time periods into a 5-min traffic flow unit for comparison. From the MAE, MAPE, and RMSE in Fig. 11, it can be seen that the two data sets have the best predictive effect at 15 min, and all three metrics have reached the lowest. With the increase of prediction time, MAE, MAPE, and RMSE gradually increase, and the prediction accuracy gradually decreases. It is worth noting that the 5-min prediction accuracy is not as good as the 15-min prediction accuracy, because prediction time is too close, and accidental factors and external factors may have greater influence, such as the influence of traffic lights. In many cities, the red light waiting time can be as long as two minutes. In the 5-min prediction, the interference of traffic lights is relatively large. In the 15-min prediction, the impact of traffic lights can be regarded as the objective rules of traffic. For the PeMS04 data set and the PeMS08 data set, the PeMS04 data set has a more complex traffic network compared with the PeMS08 data set. Thus, the prediction results of the PeMS04 data set are not as good as the PeMS08 data set, but the PeMS04 data set reflects the current traffic condition more directly. Because as the prediction time becomes longer, the accuracy of traffic flow prediction becomes lower. However, PeMS08 data set is not as obvious as the PeMS04 data set. Although the 15-min prediction result is the best, it is close to the 5-min prediction accuracy. Considering that the traffic network needs to be real time, in this article, we choose to predict the 5-min traffic flow.

Since most models are suitable for predicting the traffic flow in 1 h, in order to compare the performance, we carry out experiments to predict the traffic flow in 1 h (60 min). Followed by metrics in previous work [41], we compare the performance of ST-ChebNet and other eight models for 60 min ahead prediction on PEMS04 and PEMS08 data sets in Table I. Based on Table I, we obtain Fig. 12.

Fig. 12 shows the prediction performance of the nine models on PEMS04 and PEMS08 data sets. We can see from Fig. 12, the MAE, MAPE, and RMSE of ST-ChebNet are lower than other eight models, thus, ST-ChebNet has good performance

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

TABLE I
PREDICTION RESULTS OF THE ST-CHEBNET AND THE OTHER 8 BASELINE METHODS

Datasets	Metrics	**Baseline methods**								
		CNN	GAT	GCN	LSTM	DCRNN	STGCN	ASTGCN	Graph WaveNet	**ST-ChebNet**
PEMS04	MAE	68.10	35.08	30.89	34.13	24.70	22.70	22.93	25.45	**16.59**
	MAPE	0.92	0.34	0.28	0.33	0.17	0.15	0.17	0.17	**0.12**
	RMSE	101.25	49.32	43.52	45.31	38.12	35.55	35.22	39.70	**24.23**
PEMS08	MAE	95.56	33.56	34.68	27.21	17.68	18.02	18.61	19.13	**14.77**
	MAPE	0.8	0.21	0.22	0.16	0.11	0.11	0.13	0.13	**0.08**
	RMSE	115.18	46.20	48.59	37.64	27.83	27.83	28.16	28.16	**21.68**

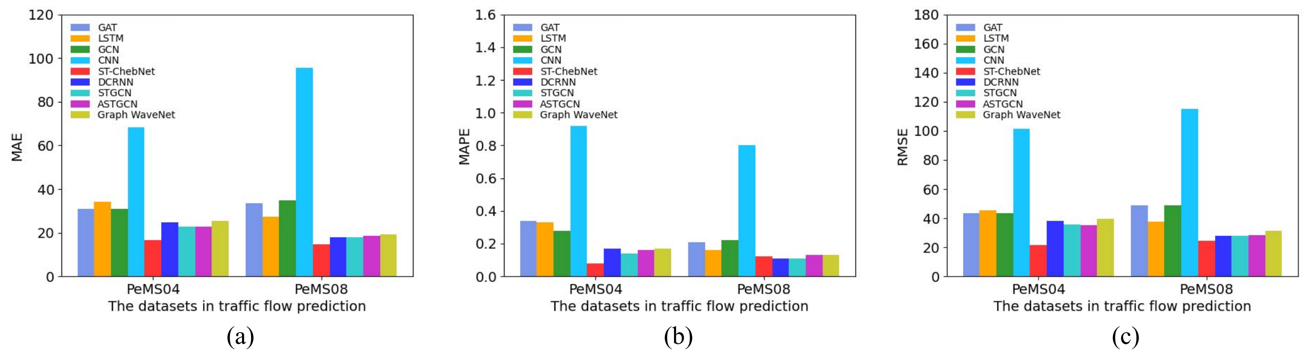


Fig. 12. Prediction results of different methods on PeMS04 and PeMS08. (a) MAE. (b) MAPE. (c) RMSE.

in traffic flow prediction. Because the complexity and parameter of ST-ChebNet are higher than other eight models, and ST-ChebNet has strong expressive ability. The K-order convolution operator of ST-ChebNet can cover the K-order neighbor nodes of the node, but other models cannot. Among these models, the three metrics of the CNN model are much higher than that of other models because of its poor ability to process the non-Euclidean data. DCRNN, STGCN, ASTGCN, and Graph WaveNet do not consider the influence of indirect neighbor nodes, so the accuracy is lower than ST-ChebNet. In summary, ST-ChebNet has great performance than other eight models in traffic flow prediction.

VI. CONCLUSION

The traffic flow prediction is a key issue, and the model required the characteristics of real time, accuracy, and adaptability. In order to achieve real time, we ensure the real-time updates of the traffic network by predicting the traffic network in the next 5 min. For accuracy, we collect traffic data through terminal equipments, such as vehicle speed sensors and roadside sensors. Then, we convert the traffic data of the urban traffic network into a graph structure. The traffic data and traffic graph for a period of time are fed into ST-ChebNet. First, the traffic data is integrated in the node through the full connection layer, and then the LSTM model extracts the node features in the time period. Finally, ST-ChebNet combines the temporal features and spatial features to obtain the prediction results. The prediction results verify the accuracy of ST-ChebNet through various evaluation metrics. In practical applications, adaptability is a very critical factor. Therefore, we have verified the adaptability of ST-ChebNet on two real-world data sets.

In this article, we mainly focus on short-term traffic flow prediction problem. As the future work, we will consider the emergency situation of special vehicles (e.g., ambulance or police car), the prediction time will be longer, and the special vehicles emergency scheme will be designed.

REFERENCES

[1] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6492–6499, Dec. 2019.
[2] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Netw.*, vol. 32, no. 4, pp. 8–14, Jul./Aug. 2018.
[3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.
[4] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic fruit classification using deep learning for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 1027–1034, Feb. 2019.
[5] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 922–929.
[6] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.
[7] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *J. Transp. Eng.*, vol. 121, no. 3, pp. 249–254, 1995.
[8] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through kalman filtering theory," *Transp. Res. B Methodol.*, vol. 18, no. 1, pp. 1–11, 1984.
[9] A. Sang and S.-Q. Li, "A predictability analysis of network traffic," *Comput. Netw.*, vol. 39, no. 4, pp. 329–345, 2002.
[10] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1700–1707, Dec. 2013.
[11] V. Lint and V. Hinsbergen, "Short-term traffic and travel time prediction models," *Artif. Intell. Appl. Crit. Transp. Issues*, vol. 22, no. 1, pp. 22–41, 2012.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.

[12] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson, "Travel time estimation for ambulances using Bayesian data augmentation," *Ann. Appl. Stat.*, vol. 7, no. 2, pp. 1139–1161, 2013.

[13] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[14] L. Mou, L. Bruzzone, and X. X. Zhu, "Learning spectral–spatial–temporal features via a recurrent convolutional neural network for change detection in multispectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 924–935, Feb. 2019.

[15] H. Peng *et al.*, "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Inf. Sci.*, vol. 521, pp. 277–290, Jun. 2020.

[16] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter-and intra-day traffic patterns," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2627–2636, May 2021.

[17] F. Zhao, G.-Q. Zeng, and K.-D. Lu, "ENLSTM-WPEO: Short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 101–113, Jan. 2020.

[18] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3913–3926, Oct. 2019.

[19] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 3529–3536.

[20] Q. Zhang, J. Chang, G. Meng, S. Xiang, and C. Pan, "Spatio-temporal graph structure learning for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 1177–1185.

[21] W. Chen, B. Liu, H. Huang, S. Guo, and Z. Zheng, "When UAV swarm meets edge-cloud computing: The QoS perspective," *IEEE Netw.*, vol. 33, no. 2, pp. 36–43, Mar./Apr. 2019.

[22] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," 2021. [Online]. Available: arXiv:2101.11174.

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," 2014. [Online]. Available: arxiv:1312.6203.

[24] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017. [Online]. Available: arXiv:1707.01926.

[25] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017. [Online]. Available: arXiv:1709.04875.

[26] L. Bai *et al.*, "STG2SEQ: Spatial–temporal graph to sequence model for multi-step passenger demand forecasting," 2019. [Online]. Available: arXiv:1905.10069.

[27] L. Zhao *et al.*, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[28] J. Zhu, Y. Song, L. Zhao, and H. Li, "A3T-GCN: Attention temporal graph convolutional network for traffic forecasting," 2020. [Online]. Available: arXiv:2006.11583.

[29] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial–temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, early access, Feb. 3, 2021, doi: 10.1109/TKDE.2021.3056502.

[30] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial–temporal graph modeling," 2019. [Online]. Available: arXiv:1906.00121.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016. [Online]. Available: arXiv:1606.09375.

[33] F. R. Chung and F. C. Graham, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.

[34] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[35] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 1944–1957, Nov. 2007.

[36] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[37] S. Guo, Y. Lin, H. Wan, X. Li and G. Cong, "Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting," in *IEEE Trans. Knowl. Data Eng.*, doi: 10.1109/TKDE.2021.3056502. '

[38] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[39] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016. [Online]. Available: arXiv:1609.02907.

[40] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017. [Online]. Available: arXiv:1710.10903.

[41] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 914–921.

[42] M. Li, P. Tong, M. Li, Z. Jin, J. Huang, and X.-S. Hua, "Traffic flow prediction with vehicle trajectories," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 294–302.

Biwei Yan received the B.S., M.S., and Ph.D. degrees in computer science and mathematical sciences from Qufu Normal University, Qufu, China, in 2015, 2017, and 2021, respectively.

She is currently a Lecturer with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong, China. Her research interests focus on blockchain, deep learning, security, and privacy.

Guijuan Wang received the B.S. and M.S. degrees in computer science from Qufu Normal University, Qufu, China, in 2013 and 2016, respectively, and the Ph.D. degree in computer science from Soochow University, Suzhou, China, in 2019.

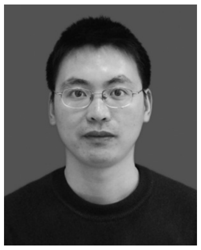
She is currently a Lecturer with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. Her research interests include parallel and distributed systems, algorithms, and interconnection architectures.

Jiguo Yu (Senior Member, IEEE) received the Ph.D. degree from the School of Mathematics, Shandong University, Jinan, China, in 2004.

He was a Full Professor with the School of Computer Science, Qufu Normal University, Qufu, Shandong, China, in 2007. He is currently a Full Professor with the Qilu University of Technology (Shandong Academy of Sciences), Jinan. He is particularly interested in designing and analyzing algorithms for many computationally hard problems in networks. His main research interests include privacy-aware computing, blockchain, intelligent IoT, cyber–physical system, distributed computing, and graph theory.

Dr. Yu is a member of ACM and a Senior Member of the China Computer Federation.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.



Xiaozheng Jin (Member, IEEE) received the B.S. degree in electromechanical engineering from Zhejiang Sci-tech University, Hangzhou, China, in 2004, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2010.

From August 2014 to August 2015, he was a Research Fellow with Rolls-Royce@NTU Corporate Laboratory, Singapore. He was a Professor with the School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, China, from September 2015 to October 2019. He is currently a Professor with the School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China, and the Shandong Key Laboratory of Computer Networks, Jinan. His current research interests include fault-tolerant control, control of multi agent systems, and security control of cyber–physical systems.

Hongliang Zhang received the B.S. degree from the Qilu University of Technology (Shandong Academy of Sciences), Jinan, China, in 2019, where he is currently pursuing the M.S. degree in computer science.

His current research interests include deep learning and blockchain.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 18:46:02 UTC from IEEE Xplore. Restrictions apply.