Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 1217588, 12 pages
https://doi.org/10.1155/2022/1217588

Research Article

AGCN-T: A Traffic Flow Prediction Model for Spatial-Temporal Network Dynamics

Jian Feng, Lang Yu, and Rui Ma

College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an 710054, China

Correspondence should be addressed to Jian Feng; fengjian@xust.edu.cn

Received 19 December 2021; Revised 9 April 2022; Accepted 12 April 2022; Published 29 May 2022

Academic Editor: Francesco Galante

Traffic prediction is the key for Intelligent Transport Systems (ITS) to achieve traffic control and traffic guidance, and the key challenge is that traffic flow has complex spatial-temporal dependence and nonlinear dynamics. Aiming at the lack of the ability to model complex and dynamic spatial-temporal dependencies in current research, this paper proposes a traffic flow prediction model Attention based Graph Convolution Network (GCN) and Transformer (AGCN-T) to model spatial-temporal network dynamics of traffic flow, which can extract dynamic spatial dependence and long-distance temporal dependence to improve the accuracy of multistep traffic prediction. AGCN-T consists of three modules. In the spatial dependency extraction module, according to the similarity of historical traffic flow sequences of different loop detectors, an adjacency matrix for the road network is constructed based on a sequence similarity calculation method, Predictive Power Score (PPS), to express latent spatial dependency; and then GCN is used on the adjacency matrix to capture the global spatial correlation and Transformer is used to capture dynamic spatial dependency from the most recently flow sequences. And then, the dynamic spatial dependency is merged with the global spatial correlation to obtain the overall spatial dependency pattern. In the temporal dependency extraction module, the temporal dependency pattern of each traffic flow sequence is learned by the temporal Transformer. The prediction module integrates both patterns to form spatial-temporal dependency patterns and performs multistep traffic flow prediction. Four sets of experiments are performed on three actual traffic datasets to show that AGCN-T can effectively capture the dynamic spatial-temporal dependency of the traffic network, and its prediction performance and efficiency are better than existing baselines. AGCN-T can effectively capture the dynamics in traffic flow. In addition to traffic flow prediction, it can also be applied to other spatial-temporal prediction tasks, such as passenger demand prediction and crowd flow prediction.

1. Introduction

Traffic flow prediction is the prerequisite for traffic control and traffic guidance, and it is also the key to the research and implementation of the Intelligent Transportation System (ITS). Traditionally, real-time traffic data are collected by in-ground loop detectors or traffic surveillance camera systems, which are fixed measure stations that obtain information in terms of flows, occupancy, speed, or videos. Prediction of one or more information of these within one or several hours ahead is typically called short-term flow prediction [1]. In this work, we focus on traffic speed prediction, which predicts the future traffic speeds for each loop detector using historical speed.

The traffic flow on a loop detector is spatially affected by the dynamic traffic of other loop detectors in the traffic network and has multiple timing characteristics such as proximity, periodicity, and trend. These complex dynamic spatial-temporal correlations make traffic flow prediction become a very challenging task. Early studies regarded traffic flow as an independent time series and used time series-based methods to predict it. The most representative one is ARIMA (Autoregressive Integrated Moving Average) model and its variants [2–4]. This type of method requires data to be stable and continuous and cannot be well adapted to dynamic and complex traffic flow prediction. Subsequent research gradually considered spatial relationships and external background data, such as locations, weather, and events, and used traditional machine learning methods for modeling [5–8]. However, these models rely heavily on expert experience in related fields and require manual feature engineering. In recent years, as deep learning has been widely used, a large number of studies have adopted RNN (Recurrent Neural Network) and/or CNN (Convolutional Neural Network) to automatically extract spatial-temporal correlation features from traffic flow, which greatly improves the prediction accuracy [9–12]. However, these works still have some limitations, including the following: ① RNN-type methods may cause serious training problems due to the vanishing gradient and explosive gradient problems; ② CNN-type methods are based on a regular network structure, but gridding the traffic network will result in the inability to effectively express non-Euclidean spatial characteristics of the traffic network. To solve the problem, the GNN (Graph Neural Networks) method, which constructs the traffic network as an irregular graph structure, has received a lot of research in recent years. In GNNs, GCN (Graph Convolutional Network) has gained the most attention because of its natural fusion of structure (spatial characteristics in traffic flow) and attributes (temporal characteristics in traffic flow) [13–17]. At the same time, research based on Graph Attention Network (GAT) is gradually increasing because the attention mechanism it uses can effectively learn the dynamics in traffic flow [18].

However, GNNs still fall short of modeling traffic flow due to the following challenges. Firstly, in the expression of spatial dynamics, existing GNNs rely heavily on a predefined network structure and lack consideration or expression of time-varying spatial correlation. However, on the one hand, existing topology construction methods for traffic networks usually consider the local connection characteristics of roads, constructing the topology based on connectivity or distance, and ignoring the global impact among roads, so the constructed traffic network may ignore the correlation among distant roads which share similar temporal patterns. On the other hand, the spatial dependence between roads is not strictly stable, and the spatial pattern of traffic flow will also change significantly over time. For example, on weekdays and weekends, or morning peak and evening peak, two specific roads usually have a different relationship. There are complex spatial dependencies between different road nodes, and the traffic flow between different areas at different times has complex temporal dependencies, so different roads have different patterns of traffic flow. Secondly, when expressing temporal dynamics, existing GNNs usually combine RNN-type methods to capture short-term trends and long-term periodicity. However, RNN-type methods have problems of time-consuming iterative propagation and gradient explosion or disappearance and cannot simulate the temporal change pattern well. Finally, the current prediction model is generally oriented to single-step prediction, which makes its forecast accuracy decrease significantly as the forecast time range increases. However, in practical applications, multistep prediction is more valuable, and how to improve the accuracy of multistep prediction becomes an urgent problem to be studied.

To meet the above challenges, we study the problem of multistep traffic prediction by proposing a new model AGCN-T (Attention-based Graph Convolution Network and Transformer) by fusing GCN and Transformer. AGCN-T extracts the global dependencies between nodes based on the similarity between time series firstly and builds the basic network topology by extracting the global dependencies among nodes in the traffic network based on the similarity among historical traffic sequences on different loops detectors. On this basis, GCN is used to capture global spatial correlation. At the same time, a spatial Transformer network with the attention mechanism learns the local influence of different loop detectors and provides the ability to dynamically perceive spatial dependent changes. And then, the temporal Transformer network is used to capture the dynamic temporal dependencies on each loop detector. Finally, the learned spatial-temporal features are fused and used for multistep traffic flow prediction. Notice that the Transformer used can obtain multistep prediction at one run instead of multiple predictions based on existing prediction results. Notice that in addition to traffic flow prediction, AGCN-T can also be applied to other spatial-temporal prediction tasks, such as passenger demand prediction and crowd flow prediction. The main contributions of the paper include the following:

(i) We adopt a network building method PPS (Predictive Power Score), to learn hidden spatial dependencies among historical traffic sequences. PPS has intuitive physical meaning. Notice the method can automatically discover invisible network structures from the data without the guidance of prior knowledge. It is the first time that PPS has been used in the construction of an adjacency matrix for a traffic network.

(ii) AGCN-T is proposed to model the dynamic spatial-temporal dependence for traffic flow prediction. To extract spatial pattern, the spatial dependency is learned by GCN and the spatial Transformer to extract the global and local spatial relationships, respectively, to indicate the periodicity and trend separately. The temporal dependency is learned through the temporal Transformer. Especially, Transformer is used instead of RNN technologies to solve the multistep prediction problem.

(iii) Experiments are conducted on real traffic data sets, and the results show that the effect of AGCN-T is better than the existing mainstream prediction methods.

The paper is organized as follows: Section 2 gives a summary of state-of-the-art research in predicting traffic flow. We then formalized the traffic prediction problem and proposed the AGCN-T model in Section 3. Section 4 discusses the experiment design and the performances of the tested models. Conclusions are then drawn in Section 5.

2. Literature Review

The field of traffic flow prediction has existed for almost five decades and covers a wide array of methodologies which can be divided into two categories, traditional statistical methods and machine learning methods.

2.1. Traditional Statistical Methods. The research field of traffic prediction has evolved greatly ever since its inception in the late 1970s, while Ahmed et al. used ARIMA for highway traffic flow prediction for the first time [2]. Subsequently, ARIMA and its variants [3, 4], VAR [19], Kalman filter [20], and other algorithms used and regarded traffic flow as time series and performed statistical analysis on historical traffic flow to realize the prediction of future ones. The advantage of these models is their simplicity, but their prediction performance is poor, and the ability to mine the potential correlations among traffic flows and various influencing factors is insufficient because most of them are based on linear assumptions.

2.2. Machine Learning Methods

2.2.1. Classical Machine Learning Methods. Due to the deficiencies of traditional statistical models, researchers flocked to machine learning models, which can learn the nonlinear relationship between traffic flow and influencing factors and greatly improve the performance of traffic flow prediction. Typical methods include Support Vector Regression (SVR) [5], Bayesian model [6], kNN [7], Random Forest Regression (RFR) [8], etc. However, machine learning models rely on artificial feature engineering rather than learning directly from raw data. Artificially defined features are often difficult to capture the overly complex spatial-temporal correlations of traffic flow, resulting in information loss. Besides, shallow and simple structures of machine learning models also limit their prediction power.

2.2.2. Deep Learning Methods. As theoretical and technological advances emerged in the middle of the 2010s, researchers started to apply DNN models for traffic prediction.

These models can automatically extract and capture the characteristics from data and have strong nonlinear data mining capabilities, showing superior performance. According to how they model spatial-temporal correlations, there are classified into temporal dependency models and spatial dependency models.

(1) Temporal Dependency Models. First of all, RNN and its variants, such as LSTM [9] or GRU [10], are neural network models that process sequential data well and are commonly used to model temporal dependency. But they only regard the traffic flow as a time series. However, these methods still treat traffic flow as time series. It is easy to lose the content learned at the previous time step, resulting in poor performance when the input data is a long series. Different from RNN, CNN is a fully convolutional network, which captures the time trend of nodes through the time convolutional layer, while the convolutional network does not rely on previous calculations, and the elements in the sequence can be parallelized. The receptive field of CNN is usually small, which is not conducive to capturing and storing long-distance dependent information [11]. Since traffic flow prediction has inherently complex spatial-temporal dependence, some research use hybrid CNN-RNN models for archiving better predictive effect than single models [12]. For example, CNN is used to extract interday and intraday traffic flow patterns, and LSTM is used to learn the evolution of intraday traffic flow in [21]. Refence [22] uses the density peak clustering algorithm and the genetic algorithm to optimize the input so that to improve the accuracy. In recent years, the self-attention mechanism adopts in traffic prediction to effectively capture long-term dependence because it can be easily adapted to data sequences of different lengths. Some recent research takes Transformer into account and achieves encouraging results [23].

(2) Spatial Dependency Models. To characterize the spatial correlation, CNN is a natural choice because it can capture the spatial local features well [11]. However, CNN is designed for the Euclidean spatial structure and needs to be based on the regular grid structure, which violates the natural non-European nature of the traffic network, so CNN cannot fully express the spatial association of traffic flow. GNNs present a new opportunity for traffic prediction due to their ability to capture spatial correlations in non-Euclidean network structures. In GNNs, GCN is used for traffic prediction in most cases. For example, T-GCN [13] is the first research to introduce GCN into traffic flow prediction; DCRNN [14] uses diffusion GCN to describe the information diffusion process in spatial networks and uses RNNs to model temporal correlation; ST-GCN [15] adopted Chebnet to capture the spatial correlation of traffic flow. These models usually combine RNN and GCN to model temporal and spatial correlation, respectively. In addition to GCN, attention-based traffic flow prediction methods gradually emerge. For example, ASTGCN [16] simultaneously employs graph convolutions and attention mechanisms to model the traffic flow; Graph WaveNet [17] proposed a new adaptive dependency matrix to capture the hidden spatial dependencies; and STGAT [18] adopted a dual path network with gating mechanisms and residual architecture, which contains gated temporal convolution and graph attention layer. However, the local similarity of road space is not considered.

These studies show that GCN is a powerful tool to deal with complex spatial-temporal dependencies, and the attention mechanism can be used to extract the dynamic changes of these dependencies. Although existing hybrid models of GNN and RNN have improved the prediction performance, there are still limitations, and before applying GNN and attention to traffic prediction tasks, there are three basic problems to be studied. Firstly, how to accurately represent the complex structure of the traffic network? Secondly, the predefined network structure is usually local and static, how to capture spatial dynamics on this basis? Thirdly, how to learn the temporal pattern of traffic flow better? Next, we will show our solutions to these problems.

3. Methodology

First, we need to formalize the traffic flow prediction problem, and then introduce the AGCN-T model, and elaborate on the details of AGCN-T in the following. A list of

Table 1: Notations in AGCN-T.

Notation	Definition and Description
$G = (V, E, A)$	Traffic network G with nodes V, edges E and adjacency matrix. A
n	Number of nodes in. V
A_{ij}	The strength of the relationship between vertex i and. j
X	Feature matrix of. G
T	Total number of historical time steps.
v_i^t	The speed detected by the loop detector i at time. t
$\hat{y}^t$	Predicted traffic speed at time. t
y^t	Observed traffic speed at time. t
θ^*	The optimal set of parameters.
X_i, X_j	Sequences of traffic detected by detectors i and. j
S_{ij}^{MAE}	MAE values of X_i predicting. X_j
S_{ij}^{M}	The median of. S_{ij}^{MAE}
$\tilde{A}$	The adjacency matrix A by adding the identity matrix. I
Z^W	Global spatial dependence matrix.
F	The dimension of the embedding vector.
A'	The normalized adjacency matrix of A'. G
D	Degree matrix of. G
S^A	The attention matrix.
Z^L	Local spatial dependence matrix.
Z^S	Spatial dependence matrix.
Z^T	Temporal dependence matrix.
Z^{ST}	Spatial-temporal dependence matrix.

abbreviations of the definitions and notations used in the paper is firstly given in Table 1.

3.1. Problem Formalization. Traffic flow prediction means to perform prediction on future traffic information, as given historical information collected by a group of loop detectors. In this paper, traffic speed adopted by loop detectors is taken as a prediction object. In this section, we first give some definitions and then formalize the traffic prediction problem.

Definition 1. (Graph for Traffic Network G): The traffic network can be abstracted as a graph, which is represented as $G = (V, E, A)$, where V is the set of vertices in graph G with $|V| = n$, represents nodes in the traffic network, such as loop detectors; E represents the set of edges between loop detectors. $A \in R^{n\times n}$ denotes the weighted adjacency matrix that is derived from the graph G. Element A_{ij} in matrix A is used to describe the relationship strength between vertex i and vertex j. Usually, the larger values mean that the two vertexes have higher correlations.

Definition 2. (Feature Matrix X): $X \in R^{T\times n}$ is the recorded historical observations, where T represents the total number of historical time steps.

$$\begin{aligned} X \in R^{T\times n} &= \left(X^1, \ldots, X^{T-1}, X^T\right)^T \\ &= \begin{bmatrix} v_1^1 & v_2^1 & \ldots & v_n^1 \\ v_1^2 & v_2^2 & \ldots & v_n^2 \\ \ldots & \ldots & \ldots & \ldots \\ v_1^T & v_2^T & \ldots & v_n^T \end{bmatrix}, \end{aligned} \tag{1}$$

where X^t is represented as the feature vector and v_i^t is the speed detected by the loop detector i at time t.

The formal definition of the traffic speed prediction problem is to find a mapping function $f = (\cdot)$, such that we can infer the snapshots of graph G with X in the future T' snapshots according to T historical observations:

$$\hat{y}^{t+T'} = f\left(G; \left(X^{t-T+1}, \ldots, X^{t-1}, X^t\right)\right). \tag{2}$$

The objective is to find parameters of the model which can minimize the error between the predicted speed and the observed ones:

$$\theta^* = \mathrm{argmin} L\left(y^{t+T\prime}, \hat{y}^{t+T\prime}; \theta^*\right), \tag{3}$$

where y^t and $\hat{y}^t$ are observed and predicated traffic at time t separately. L is the loss function and θ^* is the optimal set of parameters for the function f.

3.2. Framework of AGCN-T. Figure 1 gives an overview of AGCN-T, which contains three modules: spatial dependency extraction module, temporal dependency extraction module, and prediction module. Firstly, the spatial dependency extraction module constructs an adjacency matrix for a traffic network based on historical speed sequences of nodes in a traffic network and uses the adjacency matrix as input to mine the global spatial dependency pattern by GCN; then, the spatial Transformer is used to obtain hourly dynamic spatial dependency; secondly, the temporal dependency extraction module uses the temporal Transformer to learn temporal dependency pattern of historical traffic sequences. In the end, the prediction module integrates learned spatial and temporal dependencies and performs multistep traffic flow prediction.

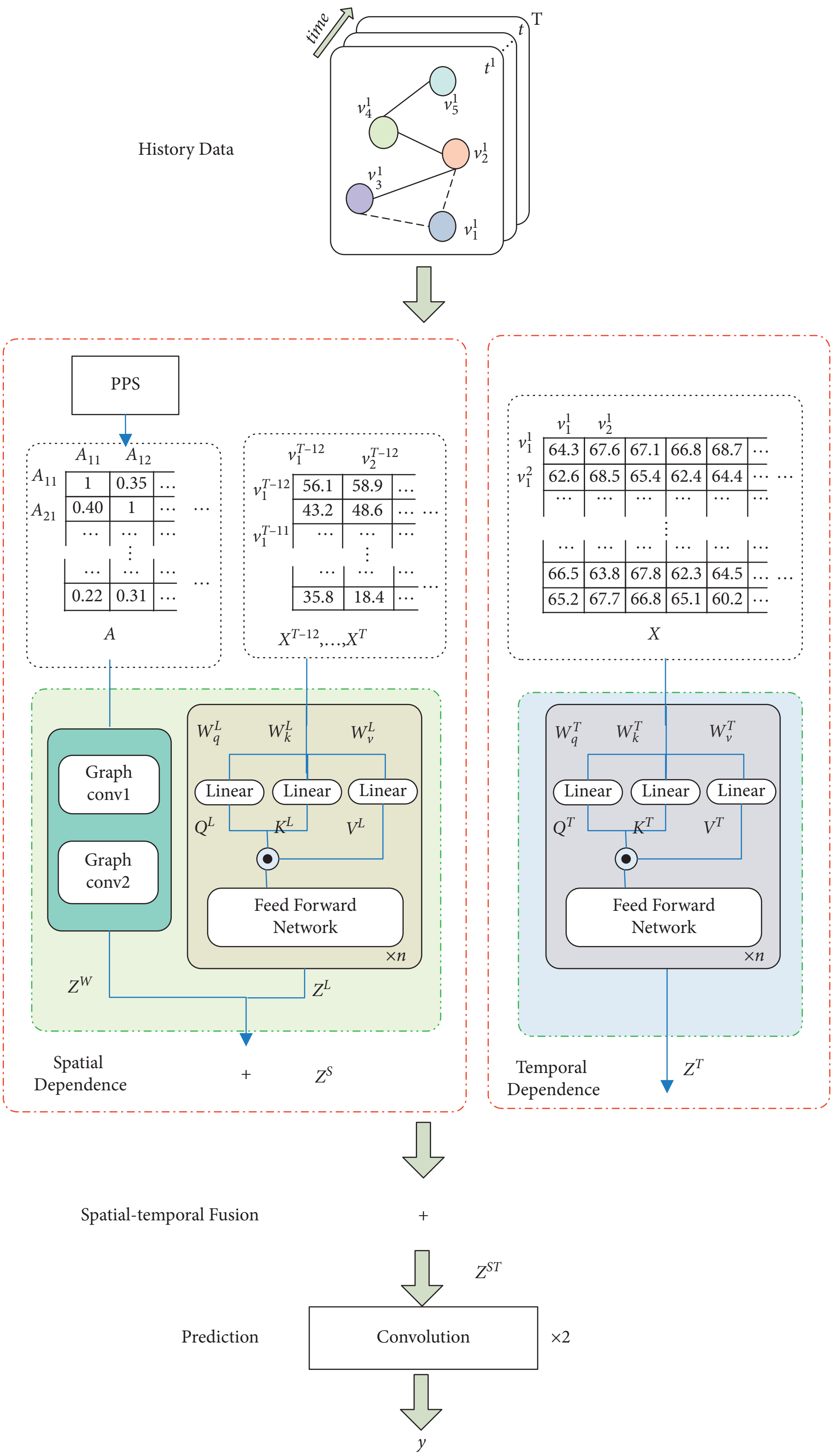


Figure 1: The framework of AGCN-T.

AGCN-T solves three difficult problems: ① adjacency matrix construction for traffic networks without connection features; ② extraction of spatial dynamic dependency; and ③ extraction of temporal dependency. In the next few sections, each part of AGCN-T is elaborated in more detail.

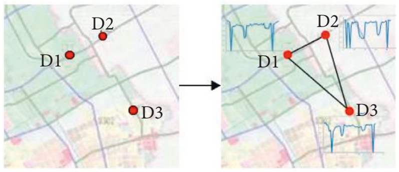
3.3. Adjacency Matrix Construction. As we have mentioned, GCN is used in AGCN-T to extract spatial correlations in the traffic network. GCN needs a predefined adjacency matrix to perform graph convolution operations, so adjacency matrix construction is the key to the success of GCN.

Current mainstream research is usually based on the physical connectivity between nodes [16] or [17] distance between latitude and longitude [24, 25] to build an adjacency matrix. The literature [26] obtains the hidden interdependence between historical traffic sequences by the Pearson correlation coefficient. However, the Pearson correlation coefficient is more suitable for expressing linear relationships, but the traffic sequences are often nonlinearly related; second, the correlation coefficient matrix is symmetric, and due to the upstream and downstream relationship between the detectors in the road network, their mutual influence is not symmetrical. To this end, AGCN-T uses the PPS method to learn the correlation between nodes. This is the first time that PPS has been used in the construction of an adjacency matrix for a traffic network.

The PPS is a normalized index (ranging from 0 to 1) that tells us how much the variable x could be used to predict the variable y. The higher the PPS index, the more the variable x is decisive in predicting the variable y. Assuming that the traffic sequences detected by the two detectors i, j in the traffic network are X_i and X_j respectively, the ability of X_i to predict X_j represents the correlation between X_i and X_j. In PPS, X_j is treated as the target variable and X_i as the only feature, and then a regression decision tree is calculated. In detail, the method needs to divide the feature space X_i. Each division examines all values of all features in X_i one by one and selects the best one as the segmentation point according to the square error minimization criterion, and then a regression decision tree is built to obtain the MAE (Mean Absolute Error) value S_{ij}^{MAE}. Record the median of S_{ij}^{MAE} is recorded as S_{ij}^{M}, then the element value of the adjacency matrix is obtained by the ability of X_i to predict X_j :

$$A_{ij} = 1 - \left(\frac{S_{ij}^{\mathrm{MAE}}}{S_{ij}^{M}}\right). \tag{4}$$

PPS models the nonlinear correlation among traffic sequences and obtains the adjacency matrix A, as shown in Figure 2.



$$A = \begin{pmatrix} 1 & 0.35 & 0.64 \\ 0.40 & 1 & 0.40 \\ 0.65 & 0.33 & 1 \end{pmatrix}$$

Figure 2: Construction of adjacency matrix by PPS.

The construction process mentioned above does not consider the influence of loop detectors themselves, so we update the adjacency matrix A by adding a self-loop.

$$\widetilde{A} = A + I, \tag{5}$$

where I is the identity matrix with size $n \times n$.

It is worth noting that in the above topology construction method, the direction dependence has been implicitly integrated into the adjacency matrix, so this paper adopts a simple and easy-to-calculate undirected graph instead of the complex directed graph.

3.4. Spatial Dependence Extraction. Traffic flows often show a multiscale correlation in spatial dimensions, including global dependence and local dependence. The traffic status in different loop detectors tends to correlate with each other, and the strength of spatial correlations varies at different locations and highly depends on the underlying traffic network structures. This is global dependence. At the same time, the traffic conditions are temporalvarying. For example, the strength of the dependence between adjacent road sections is different in the morning peak and evening peak, which means that the spatial relationship is dynamic, and this is local dependence. Establishing an extraction model for multiscale spatial dependence among nodes in the road network is the key to accurately capturing spatial dependence.

AGCN-T uses GCN to mine the global spatial dependence pattern, uses a spatial Transformer to obtain the local dynamic spatial dependence that evolves, and then fuses the learned global spatial features and local dynamic features. Combined with adjacency matrix construction, the spatial dependence extraction module can be regarded as a general message passing GNN for dynamic graph construction and feature learning.

3.4.1. Global Spatial Dependence Extraction. Taking the global correlation graph constructed by PPS as input, GCN is used to capture the inherent spatial dependence among nodes in the traffic network. The representation matrix Z^W obtained by GCN is as follows:

$$Z^W = \mathrm{GCN}(X, \widetilde{A}), \tag{6}$$

where the size of Z^W is $n \times F$, and F represents the dimension of the embedding vector. GCN is composed of two layers of convolutions:

$$\mathrm{GCN}(X, \widetilde{A}) = \mathrm{ReLU}\left(A' \mathrm{ReLU}\left(A' \mathrm{XW}_0\right) W_1\right), \tag{7}$$

where $A' = D^{-1/2}\widetilde{A}D^{-1/2}$ is the symmetric normalized adjacency matrix, in which D is the degree matrix, W_i is the weight matrix of layer i.

3.4.2. Local Spatial Dependence Extraction. After applying GCN to capture the whole spatial correlations of traffic, we are seeking a way to capture the local spatial dependence.

Here we use a spatial Transformer to capture the spatial dependence behind hourly timing information and express the dynamic impact of local changes on the traffic network structure.

The input and output of the spatial Transformer are two sequences of vectors. Notice that the input is the flow matrix of the last 12 intervals, namely, $(X^{T-12}, \ldots, X^{T-1}, X^T)^T$, with 5 minutes at intervals. In the spatial Transformer, each vector in the input sequence is linearly transformed into three vectors called query Q, key K , and value V [27]. Each output vector is computed as a weighted sum of all the values, where the weights are the outputs of a softmax layer, and the inputs of the softmax layer are scaled dot products of the corresponding query with all keys. The spatial Transformer is calculated as follows:

$$\begin{aligned} Q &= \mathrm{XW}_{q,} \\ K &= \mathrm{XW}_{k,} \\ V &= \mathrm{XW}_{v}, \\ S^A &= \operatorname{softmax}\left(\frac{\mathrm{QK}^T}{\sqrt{d_K}} V\right), \end{aligned} \tag{8}$$

where W_q, W_k, and W_v are the weight matrices corresponding to Q, K, and V respectively, S^A is the attention matrix, and d_K is the dimension of the vector K. After further learning through a three-layer feedforward neural network, the representation of local spatial dependence is obtained:

$$Z^L = \mathrm{ReLu}\left(\mathrm{ReLu}\left(S^A W_0\right) W_1\right), \tag{9}$$

where W_0, W_1 are the parameters of different layers in the feedforward neural network.

3.4.3. Fusion. The global spatial dependence learned by GCN represented by Z^W and the local spatial dependence learned by using the attention mechanism represented by Z^L are combined to adjust the spatial correlation of nodes in the graph, and thereby effectively simulate dynamic spatial dependence. Notice that the multimodal fusion mechanism of the Transformer can convert different features into a unified sequence, solve the problem of inconsistent multimodal input, and ensure the consistency of spatial-temporal feature fusion.

$$Z^S = Z^W + Z^L. \tag{10}$$

3.5. Temporal Dependence Extraction. Traffic status at the same loop detector also exhibits strong correlations over time. To capture this temporal dependence, a temporal Transformer is used to model the sequence relationship of the flow sequence. On the one hand, this is to solve the problem that the RNN-type methods cannot extract long-term periodicity well. Transformer supports the multihead self-attention mechanism instead of abandoning the recurrence mechanism in RNN, and this can avoid the “forgetting” problem and solve the challenges faced by RNN. On the other hand, the attention mechanism and position coding strategy in Transformer can dynamically capture the context-related characteristics of sequence data to realize the traffic prediction of multiple time steps better.

The calculation process of the temporal Transformer and the spatial Transformer is the same, and the difference is that the input is a series of past traffic data, that is, the historical time series feature matrix X, instead of the traffic matrix of the last 12 moments, and the output is a temporal dependence representation Z^T.

3.6. Prediction. Spatial dependence and temporal dependence affect traffic flow together. Therefore, we first merge spatial-temporal dependence learned to form spatial-temporal feature and then perform multistep prediction through two convolutional layers:

$$\begin{aligned} Z^{ST} &= Z^S + Z^T, \\ \hat{y} &= \mathrm{Conv}\left(\mathrm{Conv}\left(Z^{ST}\right)\right), \end{aligned} \tag{11}$$

where Conv $(\cdot)$ is a $1 * 1$ convolutional operation. The goal of model training is to minimize the error between the actual traffic speed y and the predicted one $\hat{y}$, so the loss function is mean absolute loss:

$$L = \| y - \hat{y} \|_1. \tag{12}$$

4. Experimental Analyses

To verify the effectiveness of AGCN-T, four sets of experiments are designed to try to answer the following questions:

(1) Question 1: How is the overall traffic prediction performance of AGCN-T as compared to various baselines?

(2) Question 2: How do the designed different submodules contribute to the model performance?

(3) Question 3: Is the PPS-based adjacency matrix construction method effective compared with the classic adjacency matrix construction method for traffic networks?

(4) Question 4: From the perspective of running time, how is the model efficiency of AGCN-T?

4.1. Experiment Preparation

4.1.1. Experimental Environment and Dataset. The experimental development environment is shown in Table 2.

Three real network-scale traffic speed datasets are utilized in the experiments.

PeMSD7 [15] collects traffic information from 228 monitoring stations in the California state highway system during the weekdays from May through June of 2012. Traffic speeds are aggregated every five minutes and normalized with Z-Score as inputs.

Table 2: Development environment.

Operating system	Processor	Memory (GB)	Development environment	Development language
Windows 10	Intel core i5-8250 CPU @ 1.60 GHz	16	PyCharm	*Python*3.5

Seattle [28] is collected from inductive loop detectors deployed on four connected freeways (I-5, I-405, I-90, and SR-520) in the Greater Seattle area and contains traffic data from 323 sensor stations over the entirety of 2015 at 5-minute intervals.

Los-loop [13] collects 207 loop detectors and their traffic speed from March 1 to March 7, 2012, on the highway of Los Angeles County. Traffic speeds are also aggregated every five minutes.

We use 60% of the data as the training set, 30% as the validation set, and 10% as the test set in strict chronological order.

4.1.2. Evaluation Standard. To summarize various evaluation indicators in the literature, the most commonly used are the following: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE), and their calculation formulas are shown below.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|.$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right|^2}. \tag{13}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{\widehat{y}_i}\right| \times 100\%.$$

where y_i and $\widehat{y}_i$ are the actual observed value and predicted one of node i respectively.

4.1.3. Baselines. We compare AGCN-T with three types of baseline methods, namely, classical statistical models such as ARIMA, DNN models that do not consider spatial information such as FC-LSTM, and GNN models based on spatial-temporal fusion such as T-GCN and ST-GCN, DCRNN.

(1) ARIMA [2] predicts future data through a combination of statistical methods including autoregression, moving average, and difference calculations on historical data.

(2) The encoder and decoder in FC-LSTM [9] both include multiple layers of LSTM and then use the fully connected layer for prediction.

(3) T-GCN [13] uses GCN to learn the spatial characteristics of the traffic network, uses GRU to learn the temporal characteristics, and then fuses them for prediction.

(4) The core of ST-GCN [15] is two ST-Conv Blocks. Each ST-Conv Block has two time-gated convolutions and a spatial GCN in between to extract temporal and spatial features, respectively.

(5) DCRNN [14] captures the spatial dependence by bidirectional diffusion random walks and captures the temporal dependence by an encoder-decoder architecture with presampling.

4.1.4. Experimental Parameter Settings. We train our model using an Adam optimizer with a learning rate of 0.0001. The dimension of AGCN-T is 512, and the number of heads in the attention of the Transformer is 8. Dropout with $P = 0.5$ is applied to the outputs of the graph convolution layer.

4.2. Evaluation Results and Analysis

4.2.1. Experiment 1: Prediction Performance of AGCN-T. Aiming to answer Question 1, Experiment 1 compares AGCN-T with baselines and predicts the speed of each node on the three data sets of PeMSD7, Seattle, and Los-loop in the future 15 minutes, 30 minutes, 45 minutes, and 60 minutes respectively, the performance of each method is indicated by MAE, MAPE, and RMSE. The experimental results are shown in Tables 3–5.

It can be seen intuitively from the figures that AGCN-T has the best performance overall, while ARIMA has the worst prediction effect. This is because ARIMA is a traditional statistical method and has limited ability to model complex traffic data with nonlinear characteristics. FC-LSTM, like ARIMA, does not consider the spatial dependence among nodes in the traffic network, so the prediction ability is lower than the other four methods which consider the spatial dependence. By performing bidirectional diffusion graph convolution on the explicitly designed directed matrix to consider the influence of directionality, DCRNN outperforms ST-GCN and T-GCN. However, the influence of directionality is complicated and hard to measure. Instead, AGCN-T uses PPS to capture the dynamical directed spatial dependence in a data-driven manner to achieve similar results with a simple undirected adjacent matrix, and adopts Transformer to enhance the dynamical spatial dependence, demonstrating the effectiveness of modeling. Furthermore, compared with ST-GCN adopting three pairs of spatial and temporal units, AGCN-T consistently outperforms it with the combination of GCN and Transformer.

4.2.2. Experiment 2: Sub-Module Comparison. For Question 2, to evaluate the effects of modules including temporal dependence extraction, spatial-dependence extraction, and spatial-temporal fusion in AGCN-T, disassemble AGCN-T to form four methods: AGCN-T-T, AGCN-T-A, AGCN-T-G, and AGCN-T, while AGCN-T-T only implements the temporal dependence extraction module in AGCN-T,

Table 3: Prediction performance of each method on PeMSD7.

Time (min)	Evaluation Metrics	PeMSD7					
		ARIMA	FC-LSTM	T-GCN	ST-GCN	DCRNN	AGCN-T
15	MAE	9.3507	9.0169	6.4001	6.3835	6.1991	**4.3097**
	MAPE (%)	12.6691	12.0103	10.5345	10.3706	10.2325	**10.0922**
	RMSE	9.9915	9.8204	7.3981	6.8004	6.1723	**4.8063**
30	MAE	9.1279	9.4351	7.2408	6.2026	6.2422	**4.2688**
	MAPE (%)	13.3908	12.4609	11.2786	10.5237	10.3404	**10.1716**
	RMSE	9.7209	9.6602	7.7804	6.6337	6.6007	**4.7951**
45	MAE	9.3672	9.1027	7.4326	7.0218	6.4205	**4.2936**
	MAPE (%)	13.0813	12.9773	11.3014	11.1254	10.1624	**10.0164**
	RMSE	10.8952	9.6274	8.1323	7.3609	6.9523	**4.7809**
60	MAE	10.1145	9.2512	8.1004	7.3711	7.0213	**4.4184**
	MAPE (%)	13.8922	12.6821	11.2004	11.1055	11.3105	**10.3243**
	RMSE	10.9011	9.8308	8.5835	7.6606	8.9028	**4.9245**

Table 4: Prediction performance of each method on Seattle.

Time (min)	Evaluation Metrics	Seattle					
		ARIMA	FC-LSTM	T-GCN	ST-GCN	DCRNN	AGCN-T
15	MAE	8.5586	7.3821	6.2605	6.1344	6.1285	**4.2835**
	MAPE (%)	15.3808	14.6208	13.4823	12.3895	**12.0035**	12.5774
	RMSE	8.8708	8.4208	6.7457	6.6926	6.2604	**4.6178**
30	MAE	8.4039	7.4755	6.3435	6.6204	6.4912	**4.3161**
	MAPE (%)	16.2407	15.5821	14.6907	14.1455	12.4211	**12.0112**
	RMSE	8.7053	7.8238	6.7307	6.9121	6.8134	**4.7331**
45	MAE	9.3711	8.1826	6.3942	6.0143	6.2233	**4.2224**
	MAPE (%)	16.7621	15.6608	13.3916	13.2402	13.1903	**11.9529**
	RMSE	9.6308	9.6313	6.9763	6.6647	6.7311	**4.6066**
60	MAE	9.3907	8.0623	7.0564	7.2304	7.4671	**4.4373**
	MAPE (%)	16.7206	15.3104	14.4216	13.1615	13.5909	**12.4127**
	RMSE	10.3506	9.1268	8.4527	7.9032	7.9124	**4.9595**

Table 5: Prediction performance of each method on Los-loop.

Time (min)	Evaluation Metrics	Los-loop					
		ARIMA	FC-LSTM	T-GCN	ST-GCN	DCRNN	AGCN-T
15	MAE	8.6587	7.5821	6.5607	6.0342	6.3283	**5.5518**
	MAPE (%)	15.9802	14.9203	15.9824	14.6891	**14.0238**	15.2011
	RMSE	8.3708	8.9205	8.2458	8.9921	8.5604	**8.1386**
30	MAE	8.9032	7.9756	6.7432	6.5203	6.0904	**5.4398**
	MAPE (%)	16.3402	15.9821	15.9903	14.3457	**14.0214**	15.0389
	RMSE	8.2057	7.9231	8.2459	**8.0112**	8.2137	8.0911
45	MAE	9.8713	8.9825	6.7943	6.0346	6.9239	**5.4472**
	MAPE (%)	16.8629	15.7609	15.5914	15.3409	**14.5908**	15.2411
	RMSE	9.4309	9.4312	8.9403	8.7621	8.9316	**8.1582**
60	MAE	9.9908	8.4623	7.3568	7.0309	7.4451	**5.5311**
	MAPE (%)	16.9203	15.3102	15.9213	15.2652	15.5809	**15.0778**
	RMSE	10.4507	9.0238	9.9522	8.7012	8.9128	**8.1084**

AGCN-T-A implements the temporal extraction module and the global spatial dependence extraction submodule, and AGCN-T-G implements the temporal extraction module and the local spatial dependence extraction submodule. AGCN-T contains all modules. Tables 6–8 show the prediction performance of each method.

Observing Tables 6–8, it can be seen that with the increase in prediction time, the overall performance of AGCN-T is always better than the other three variants, which shows that the comprehensive use of all modules can achieve the best prediction effect. At the same time, the prediction performance of AGCN-T-G and AGCN-T-A is better than AGCN-T-T, indicating the importance of considering spatial dependence extraction; the overall prediction effect of AGCN-T-G and AGCN-T-A is similar, while AGCN-T-A is better on PeMSD7 and AGCN-T-G is slightly better on Los-

Table 6: Performance of variants of AGCN-T on PeMSD7.

Time (min)	Evaluation metrics	AGCN-T-T	AGCN-T-A	AGCN-T-G	AGCN-T
15	MAE	5.6168	4.5984	4.7472	**4.4256**
	MAPE (%)	14.2441	11.3237	11.8711	**10.8202**
	RMSE	8.3871	7.1304	**5.7158**	6.7791
30	MAE	5.6118	4.5819	4.7474	**4.4363**
	MAPE (%)	14.1983	11.2825	11.7499	**10.7861**
	RMSE	8.4001	7.2551	5.8557	**5.5786**
45	MAE	5.6667	5.6095	**4.7581**	6.9967
	MAPE (%)	14.3808	11.7872	11.8599	**10.8902**
	RMSE	8.4702	7.3045	**6.8202**	6.8397
60	MAE	5.7903	**4.7118**	5.2767	6.8753
	MAPE (%)	14.6821	11.9023	12.9301	**11.7486**
	RMSE	8.6377	7.7448	7.8442	**7.1922**

Table 7: Performance of variants of AGCN-T on Seattle.

Time (min)	Evaluation metrics	AGCN-T-T	AGCN-T-A	AGCN-T-G	AGCN-T
15	MAE	5.4521	4.2821	4.5121	**3.9826**
	MAPE (%)	17.4125	12.3318	13.5921	**11.1394**
	RMSE	8.2782	6.3716	6.8016	**5.9907**
30	MAE	5.4356	4.2468	4.5951	**4.0206**
	MAPE (%)	17.8181	12.1546	13.4543	**11.2235**
	RMSE	8.2469	6.3168	6.8143	**6.0103**
45	MAE	5.4483	4.2385	4.6619	**4.2175**
	MAPE (%)	17.8553	12.1304	13.6041	**11.2301**
	RMSE	8.2661	6.3378	6.8343	**6.1332**
60	MAE	5.5267	4.3451	4.7084	**4.3032**
	MAPE (%)	18.0679	12.4229	13.9697	**11.6223**
	RMSE	8.3715	6.4904	7.0653	**6.1932**

Table 8: Performance of variants of AGCN-T on Los-loop.

Time (min)	Evaluation metrics	AGCN-T-T	AGCN-T-A	AGCN-T-G	AGCN-T
15	MAE	6.4086	5.6626	**5.4912**	5.5518
	MAPE (%)	17.8962	15.5348	15.5348	**15.2011**
	RMSE	9.4501	8.3365	8.2559	**8.0986**
30	MAE	6.4682	5.6977	5.4644	**5.4398**
	MAPE (%)	18.0092	15.6929	15.6609	**15.0389**
	RMSE	9.5057	8.3926	8.2992	**8.1511**
45	MAE	6.4774	5.7609	5.5355	**5.4472**
	MAPE (%)	18.0913	15.8714	15.6214	**15.2411**
	RMSE	9.5265	8.4556	8.4143	**8.1982**
60	MAE	6.5484	5.7511	5.6634	**5.5311**
	MAPE (%)	18.3326	16.8024	15.9994	**15.0778**
	RMSE	9.6421	8.5178	8.4268	**8.2084**

loop, indicating that for traffic flow prediction, both local and global spatial dependencies are valuable.

4.2.3. Experiment 3: Effect of PPS. In the construction of the adjacency matrix, AGCN-T uses PPS instead of the distance between loop detectors (a.k.a. Adj) or Pearson correlation between traffic sequences (a.k.a. Cov). To answer Question 3 and to study the influence of different spatial structure representation methods on the prediction effect, Experiment 3 verifies the effect of the PPS by comparing the prediction effect through changing the input of GCN in AGCN-T to the adjacency matrix constructed by Adj, Cov and PPS, respectively. The prediction results are shown in Table 9.

It can be seen from Table 9 that after the three data sets of PeMSD7, Seattle, and Los-loop are verified that the adjacency matrix constructed by PPS enables AGCN-T to achieve the best performance. The adjacency matrix constructed based on the distance of the latitude-longitude pair is only suitable for situations where there is a fixed spatial-temporal relationship between loop detectors and cannot reflect the dynamic influences of traffic flows. On the contrary, Pearson correlation and PPS are suitable for expressing the dynamic influence among loop detectors

Table 9: Effect of PPS.

Time (min)	Evaluation Metrics	PeMSD7			Seattle			Los-loop		
		Adj	Cov	PPS	Adj	Cov	PPS	Adj	Cov	PPS
15	MAE	5.5110	4.5991	**4.3097**	6.9075	5.6525	**5.1017**	5.6016	4.6670	**4.2835**
	MAPE (%)	13.7036	11.2313	**10.4292**	18.9328	15.5131	**13.8375**	18.1431	14.1301	**12.5277**
	RMSE	6.0454	5.4760	**4.8063**	7.2819	6.0676	**5.9637**	7.0657	5.8907	**4.8178**
30	MAE	5.4933	4.6316	**4.2688**	6.8810	5.5719	**5.1636**	5.6003	4.6786	**4.3160**
	MAPE (%)	13.6225	11.1215	**10.3271**	18.9312	15.2345	**13.8739**	18.3123	13.9333	**12.1341**
	RMSE	6.0435	5.4896	**4.7950**	7.2789	6.0461	**5.9858**	7.0712	5.9038	**4.8330**
45	MAE	5.6563	4.5968	**4.2936**	6.9114	5.7559	**5.0268**	5.6209	4.7169	**4.2224**
	MAPE (%)	14.1132	10.9292	**10.4136**	18.9304	15.2732	**13.5733**	18.2301	13.7352	**11.9352**
	RMSE	6.0765	5.4725	**4.8009**	7.2886	6.1116	**5.9479**	7.0676	5.9109	**4.8066**
60	MAE	5.7055	4.6153	**4.4184**	6.9624	5.8387	**5.2681**	5.6838	4.7478	**4.4373**
	MAPE (%)	14.0915	11.3228	**10.8241**	19.1303	15.7121	**14.1711**	18.3551	14.1812	**12.4122**
	RMSE	6.0890	5.4687	**4.8245**	7.2997	6.1240	**6.0076**	7.0817	5.9061	**4.8595**

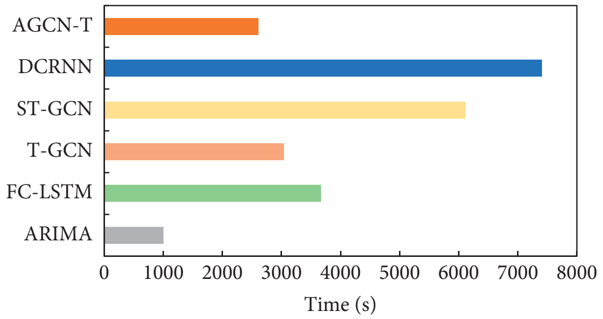


Figure 3: Comparison of efficiency.

because they can reflect joint change degree of speed among detectors. However, Pearson correlation tends to extract symmetrical and linear influences, which do not conform to the asymmetric and nonlinear nature of traffic flow. PPS reflects the ability of one node to predict another node, which is not affected by symmetry requirements and has no linear correlation restrictions, so the constructed adjacency matrix is more in line with the characteristics of traffic flow.

4.2.4. Experiment 4: Comparison of Efficiency. Aiming to Question 4, we compare the computation cost of AGCN-T with its baselines. Figure 3 gives the running time of each method on PeMSD7.

As can be seen from Figure 3, due to the use of the statistical method, the efficiency of ARIMA is significantly higher than that of deep learning-based methods and GNN-based methods. Compared with ST-GCN and DCRNN, AGCN-T has doubled its operating efficiency. This is because AGCN-T generates multistep predictions in one run while the other two have to produce the results conditioned on previous predictions. FC-LSTM and T-GCN adopt RNN technology which requires iterative training and learning, so their computational efficiency is lower than that of Transformer, which calculates multistep predictions in one run.

In summary, after the comparison of the above four sets of experiments, AGCN-T proposed in the paper has a certain improvement in the multistep traffic prediction, and also has obvious advantages in prediction efficiency.

5. Conclusions

In this paper, a novel traffic prediction model AGCN-T is proposed. A matrix construction method PPS is used to identify the asymmetric and nonlinear relationship in data sets. AGCN-T can dynamically model spatial dependency by GCN and spatial Transformer and temporal dependency by temporal Transformer. Four sets of experiments on three real datasets demonstrate the superior performance of AGCN-T in multistep prediction. Based on AGCN-T, an experimental platform for traffic flow prediction is realized, and it can be used in practice after expansion and optimization.

However, AGCN-T only considers the traffic data itself, ignoring the impact of other related information such as traffic events such as accidents [29] and so on. Multisource data will be the direction of data sources for traffic flow prediction in the future. How to effectively use these data for more accurate modeling will be continually studied in the follow-up work for traffic prediction on multisource data.

Data Availability

The data used are openly available at https://github.com/I-am-YuLang/AGCN-T.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Shaanxi Provincial Natural Science Foundation Project (No. 2020JM-533).

References

[1] L. N. N. Do, H. L. Vu, B. Q. Vo, Z. Liu, and D. Phung, “An effective spatial-temporal attention based neural network for traffic flow prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 108, no. 9, pp. 12–28, 2019.

[2] M. S. Ahmed and A. R. Cook, “Analysis of freeway traffic time-series data by using Box-Jenkins techniques,”

Transportation Research Record Journal of the Transportation Research Board, vol. 773, no. 722, pp. 1–9, 1979.

[3] B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results,” *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[4] L. N. Zhao, X. Y. Wen, Y. P. Wang, and Y. M. Shao, “A novel hybrid model of ARIMA-MCC and CKDE-GARCH for urban short-term traffic flow prediction,” *IET Intelligent Transport Systems*, vol. 129, no. 6, pp. 206–217, 2022.

[5] L. Cai, Q. Chen, W. H. Cai, X. M. Xu, T. Zhou, and J. Qin, “SVRGSA: a hybrid learning based model for short-term traffic flow forecasting,” *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1348–1355, 2019.

[6] D. Wang, C. Wang, J. Xiao, Z. Xiao, W. Chen, and V. Havyarimana, “Bayesian optimization of support vector machine for regression prediction of short term traffic flow,” *Intelligent Data Analysis*, vol. 23, no. 2, pp. 481–497, 2019.

[7] L. J. Yang, Q. Yang, Y. H. Li, and Y. Q. Feng, “K - Nearest Neighbor model based short-term traffic flow prediction method,” in *Proceedings of the 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science*, pp. 27–30, Wuhan, China, December2019.

[8] L. Zhang, N. R. Alharbe, G. Luo, Z. Yao, and Y. Li, “A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction,” *Tsinghua Science and Technology*, vol. 23, no. 4, pp. 479–492, 2018.

[9] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems*, pp. 3104–3112, Morgan Kaufmann, Montreal Canada, December 2014.

[10] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, Wuhan, China, January 2017.

[11] X. Ma, H. Yu, Y. Wang, and Y. Wang, “Large-scale transportation network congestion evolution prediction using deep learning theory,” *PLoS One*, vol. 10, no. 3, Article ID e0119044, 2015.

[12] Y. Li, S. Chai, Z. Ma, and G. Wang, “A hybrid deep learning framework for long-term traffic flow prediction,” *IEEE Access*, vol. 9, Article ID 11271, 2021.

[13] L. Zhao, Y. Song, C. Zhang et al., “T-GCN: a temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[14] Y. G. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: data-driven traffic forecasting,” in *Proceedings of the 6th International Conference on Learning Representations*, pp. 678–685, ICLR, Vancouver Canada, April 2018.

[15] B. Yu, H. T. Yin, and Z. X. Zhu, “Spatio temporal graph convolutional networks: a deep learning framework for traffic forecasting,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, Morgan Kaufmann, Stockholm Sweden, July 2018.

[16] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 922–929, AAAI, Macao, China, February 2019.

[17] Z. H. Wu, S. R. Pan, G. Long, J. Jiang, and C. Zhang, “Graph Wavenet for deep spatial-temporal graph modelling,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, Morgan Kaufmann, Macao China, August 2019.

[18] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, “STGAT: spatial-temporal graph attention networks for traffic flow forecasting,” *IEEE Access*, vol. 8, Article ID 134372, 2020.

[19] E. Zivot and J. H. Wang, “Vector Autoregressive Models for Multivariate Time Series,” *Modeling Financial Time Series with S-Plus®*, pp. 369–413, Springer, Berlin, Germany, 2016.

[20] J. Nie, “Real time prediction method of urban dynamic traffic data flow based on Kalman filter,” *Advances in Transportation Studies*, vol. 2021, pp. 33–42, 2021.

[21] D. F. 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 Transactions on Intelligent Transportation Systems*, vol. 22, pp. 1–10, 2020.

[22] D. Ma, X. B. Song, J. Zhu, and W. Ma, “Input data selection for daily traffic flow forecasting through contextual mining and intra-day pattern recognition,” *Expert Systems with Applications*, vol. 176, Article ID 114902, 2021.

[23] H. Yan, X. Ma, Z. Pu, and Z. Y. Pu, “Learning dynamic and hierarchical traffic spatiotemporal features with Transformer,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, pp. 1–14, 2021.

[24] L. Bai, L. Yao, and S. S. Kanhere, “STG2Seq: spatial-temporal graph to sequence model for multistep passenger demand forecasting,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1981–1987, Morgan Kaufmann, Macao, China, August 2019.

[25] J. J. Q. Yu and J. Gu, “Real-time traffic speed estimation with graph convolutional generative autoencoder,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, 2019.

[26] D. Chai, L. Y. Wang, and Q. Yang, “Bike flow prediction with multi-graph convolutional networks,” in *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, pp. 397–400, ACM, Seattle, USA, November 2018.

[27] A. Vaswani, N. Shazeer, and N. Parmar, “Attention is all you need,” in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, Long Beach, CA, USA, November 2017.

[28] Z. Cui, R. Ke, Z. Pu, X. Ma, and Y. Wang, “Learning traffic as a graph: a gated graph wavelet recurrent neural network for network-scale traffic prediction,” *Transportation Research Part C: Emerging Technologies*, vol. 115, Article ID 102620, 2020.

[29] B. B. Wang, Y. F. Lin, S. G. Guo, and H. Y. Wan, “GSNet: learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting,” in *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI 2021)*, pp. 4402–4409, Vancouver, Canada, February 2021.