Hindawi
Journal of Mathematics
Volume 2023, Article ID 6933344, 15 pages
https://doi.org/10.1155/2023/6933344

Research Article

Attention-Based Gated Recurrent Graph Convolutional Network for Short-Term Traffic Flow Forecasting

Ping Lou, Zihao Wu, Jiwei Hu, Quan Liu, and Qin Wei

School of Information Engineering, Wuhan University of Technology, Wuhan, Huber 430070, China

Correspondence should be addressed to Jiwei Hu; hujiwei@whut.edu.cn

Received 6 October 2022; Revised 7 November 2022; Accepted 31 March 2023; Published 19 April 2023

Academic Editor: Sheng Du

Traffic flow prediction is the basis of dynamic strategies and applications of intelligent transportation systems (ITS). Accurate traffic flow prediction is of great practical significance in alleviating road congestion and reducing urban road traffic safety hazards. It is challenging since the traffic flow has highly non-linear and complex patterns due to external factors such as time and space. Due to the high stochasticity and uncertainty of traffic flow, the difficulty of traffic flow prediction increases gradually with increasing time steps. The prediction performance of most existing short-term traffic flow prediction methods deteriorates rapidly for longer time steps. In addition, different methods are compared on the same time-granularity dataset, leaving the adaptability and robustness of these methods undervalidated. To address the above challenges, a new traffic forecasting method, named Attention-Based Gated Recurrent Graph Convolutional Network (AGRGCN) is presented for short-term traffic flow prediction. The method can extract spatialtemporal dependencies in traffic flow. In addition, an attention mechanism, which can adaptively capture traffic data relationships at different time steps, is introduced to alleviate the problem of faster deterioration of model prediction performance for longer time steps. Using a road network distance-based graph enables the method better to capture the topological information in traffic flow data. Experiments were conducted on two traffic datasets with different time granularity to predict traffic flow in highway and urban contexts. The experimental results show that our model has certain advantages.

1. Introduction

Recently, with the rapid development of urbanization and the increasing population and vehicles, the urban transportation system is facing great pressure. Fortunately, with advances in the Internet of Things and urban computing, more and more sensors are distributed across urban road networks, making real-time acquisition and analysis of vast amounts of traffic data possible. Traffic data involve traffic speed or flow, which can reflect the current traffic state and be used to predict future traffic conditions. Among those tasks, traffic flow prediction is an indispensable part of the Intelligent Transportation System [1] (ITS) and it is the basis of the dynamic strategy and application of ITS [2]. Accurate traffic flow prediction is of great practical significance in alleviating road congestion and reducing urban road traffic safety hazards. In addition, it also helps people to plan their daily trips, thus, improving their quality of life.

Traffic flow prediction is a typical problem about spatialtemporal data forecasting. Traffic data are continuously collected by sensors at fixed locations on traffic roads at a certain frequency. There is a strong spatialtemporal correlation of traffic flows among nodes on the traffic road network. Therefore, it is a very challenging issue on how to extract potential patterns and features from these complex and nonlinear spatialtemporal traffic data and make an accurate prediction of the traffic road conditions in the future period.

Traffic flow prediction has been extensively studied in the past decades. It originated from traditional statistical-based methods such as historical averaging (HA), autoregressive integrated mean shift (ARIMA) [3], Kalman filter [4], autovector regression (VAR) [5], k-nearest neighbor model [6], and Bayesian model [7]. However, these traditional methods often need to rely on the assumption of data smoothness, which cannot address the uncertainty and nonlinear characteristics of dynamic traffic flow and cannot cope with the unexpected events in the traffic system.

Later, deep learning-based methods such as s long short-term memory network (LSTM) [8, 9], Gated circulation unit (GRU) [10], and convolutional neural Network (CNN) were used for traffic flow prediction and showed better performance because they are able to extract temporal or spatial dependencies in traffic flow. Initially, grid-based graphs were applied to represent spatial dependencies in traffic flows [11–13] and then these graphs were fed to CNN to extract temporal dependencies. However, these CNN-based methods are not capable of handling non-Euclidean distance structure data and have some limitations in capturing spatial dependencies [14].

Figure 1 illustrates the non-linear relationship between the traffic flow and the geographical location of each node in the traffic network. It is obvious from Figure 1(b) that the traffic flow at node A has a clear morning and evening peak feature, while there is only one peak at node B and node C. Obviously the trend of traffic flow curves of node A and node C, which is closer in Euclidean distance as shown in Figure 1(a), are less coincident than those of nodes B and C. This example illustrates that the spatial dependencies among traffic flows depends mainly on the location distribution of sensors and geographic information, which are determined by the topological information among nodes in the road network rather than the Euclidean distance.

In recent years, graph convolutional network (GCN) stand out for their ability to handle non-Euclidean distance structure data [11, 15, 16] and have been increasingly noticed by scholars and applied to extract potential spatialtemporal properties in traffic flows [17–21]. For example, the diffusion convolutional recurrent neural network (DCRNN) proposed by Li et al. [22] adopted a sequence-to-sequence (Seq2Seq) model to extract spatialtemporal dependencies from past traffic flows for traffic prediction. Still, its ability to capture temporal dependencies in traffic flows is insufficient, and as a result, its prediction performance deteriorates rapidly for longer time steps.

In order to solve the appealing problem, an attention-based graph convolutional gated recurrent network, namely, AGRGCN, is presented for traffic flow prediction, which is able to efficiently capture the underlying patterns and temporal correlations in traffic flows. This model is able to adaptively capture the relationship among traffic flows at different time steps and still maintain good performance on multistep prediction.

The main contributions are as follows:

(1) A sequence-to-sequence (Seq2Seq) framework, named attention-based spatialtemporal graph convolution network (AGRGCN), is proposed for traffic flow prediction, which can effectively extract the spatial and temporal features of traffic flow.

(2) To alleviate the problem of rapid deterioration of model performance for longer time steps, an attention mechanism is introduced to adaptively capture traffic data relationships at different time steps.

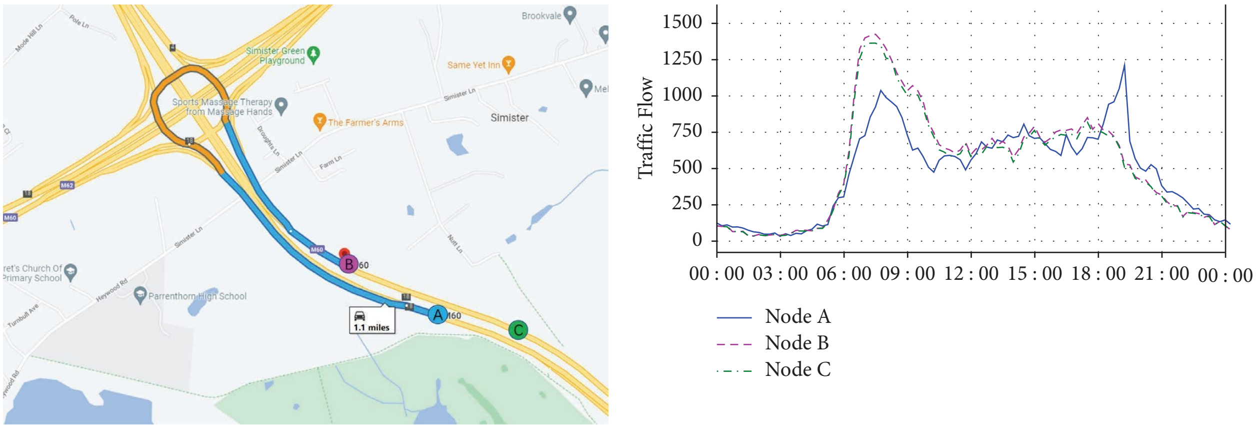
(3) Experiments are conducted on two traffic datasets with different time-granularity to validate the adaptability and robustness of AGRGCN. Compared with the baseline methods, our model performs better and is less time-consuming.

2. Related Work

Traffic flow prediction has been extensively studied by many scholars for many years. These scholars put forward many effective models and methods according to different application scenarios, which could be divided into the following three categories:

2.1. Consideration of Temporal Dependencies. Initially, scholars focused only on the temporal dependencies of traffic flows. Some classical time series models, such as HA, ARIMA, and VAR, were used to extract the temporal dependencies in traffic flows. In 1995, Hamed et al. [23] used an ARIMA model for urban traffic flow prediction. In 2003, Williams et al. [24] proposed a model of seasonal ARIMA and used it on two real ITS data sets and conducted experiments, which yielded empirical results consistent with the theoretical assumptions. Chuwang and Chen et al. [25] used an ARIMA model to implement the prediction of daily and weekly passenger demand, respectively, and experimentally derived the ARIMA model parameter settings for the two tasks. However, the ARIMA-based approach requires the original data to satisfy the assumption of smoothness, which is difficult to apply to non-linear, complex, and variable traffic road conditions. Some traditional machine learning methods were used for traffic flow prediction to fully exploit the time dependencies in traffic data. For instance, in 2004, Wu et al. [26] used SVR for travel time prediction and demonstrated the applicability and good performance of support vector regression in traffic data analysis. In 2013, Li et al. [27] used a Gaussian loss function support vector regression (SVR) model (Gaussian-SVR) for urban traffic flow prediction to reduce the random error of traffic flow data series and achieve better prediction results. Yang et al. [28] used a fuzzy-based SVR (FSVR) model to solve the international airport cargo volume forecasting problem. The ability to handle the uncertainty and imprecision of time series by using fuzzy sets improves the prediction accuracy of the overall time series model.

As deep learning continues to make breakthroughs in learning tasks such as natural language processing and computer vision, scholars have started to study on how to apply deep-learning techniques to traffic flow prediction tasks. For example, Wang [29] used multilayer perceptron (MLP) to study the prediction of short-time traffic flow on highways. In order to capture the time dependencies in traffic flow more accurately, some scholars have used variants of recurrent neural networks (RNN) time series models and their variants (LSTM [8] and GRU [13]) for traffic flow prediction with good performance. Crivellari and Beinat [30] proposed a multiobjective LSTM-based neural network regulator to predict spatially distributed urban traffic. Zhao



(a)

(b)

Figure 1: Nonlinear relationship between spatial dependence and Euclidean distance: (a) location distribution of nodes in the traffic network; (b) traffic curve of different nodes in a day.

et al. [31] built a cascaded LSTM network and integrated the origin destination correlation (ODC) matrix representing spatialtemporal correlations into the proposed network.

However, such methods only consider the temporal dependencies in traffic flow and ignore the spatial dependencies among different nodes in the traffic network.

2.2. Extracting Spatialtemporal Dependencies with CNN.

In order to capture the temporal dependencies and spatial dependencies in traffic flow, many scholars have modeled road traffic networks, using convolutional neural networks (CNNs) to extract the spatial dependencies in traffic flow and to extract the temporal dependencies in traffic flow to achieve more accurate traffic flow prediction. For example, Zhang et al. [32] divided the urban area into equal-sized grids, constructed a grid-based traffic network map, and designed a deep spatialtemporal residual network ST-ResNet to predict traffic flow. Jin et al. [33], inspired by the ST-ResNet model, constructed spatialtemporal recurrent convolutional networks (STRCNs) model, which combines CNN with LSTM for capturing the spatialtemporal dependencies of regional traffic, achieved better prediction results than the ST-ResNet model. Although these models can reasonably consider the spatialtemporal correlations among traffic flows in various city regions and extract rich spatialtemporal features, they can only handle Euclidean structured data and are inadaptive to non-Euclidean structured data. Crivellari and Beinat [30] proposed an LSTM-based method so as to predict urban traffic flows distributed over multiple reference locations in the city. Bai et al. [34] proposed a multitask convolutional recurrent neural network (MT-CRNN) framework that combines CNN and LSTM and mixes external features together, such as season, temperature, and air quality, to predict passengers' demand for multiple features from different domains.

Although these models can consider the spatialtemporal relationships among nodes in a traffic network through a grid-based graph, they are unable to adapt non-Euclidean structured data, which has some limitations.

2.3. Extracting Spatialtemporal Dependencies with GCN.

Later, some scholars investigated how to apply graph convolution techniques for spatial-temporal data mining. Zhang et al. [21] proposed a hybrid graph convolutional network HGCN to predict traffic flow at highway toll booths, which considered both spatial-temporal and external factors, including weather conditions and date types. Zhao et al. [35] proposed a temporal graph convolutional network (T-GCN) model which combines graph convolution with GRU to capture the spatialtemporal correlation of traffic flow. Li et al. [22] fused diffusion convolution with GRU to capture the spatialtemporal dependencies of traffic flow, designed a DCRNN model, and performed multistep traffic flow prediction based on the encoder-decoder framework. Yu et al. [36] designed a spatialtemporal graph convolution model (STGCN) by using graph convolution and gated convolution to capture the spatialtemporal dependencies of vehicle speed on each road segment on the highway, respectively, which outperforms DCRNN and the model training time cost was greatly reduced. Later, Diao et al. [37] improved Yu's work by designing a Laplacian matrix estimator and proposed a dynamic graph convolutional neural network model (DGCNN). Guo et al. [17] modeled the correlation among the target traffic to be predicted and its recent traffic, daily cycle traffic, and weekly cycle traffic, and introduced a spatial-temporal attention mechanism for capturing the spatial-temporal correlation among nodes. Bai et al. [38] proposed the adaptive graph convolution recurrent network (AGCRN) by adaptively constructing a traffic network graph through a learnable adjacency matrix to capture the spatial-temporal correlation of traffic data, and the multistep prediction performance of the model deteriorates slowly.

However, most of the existing GCN-based prediction methods have some performance limitations and poor robustness in multipart prediction.

3. Preliminaries

In this section, traffic flow prediction issues and several key concepts will be described as follows:

Traffic flow prediction is a typical time-series data prediction problem, which aims to predict future road conditions in a certain period based on the previously observed traffic condition. Generally, traffic flow prediction can be divided into short-term (less than 60 minutes) flow prediction and long-term (over 1 hour) prediction. The former is of greater practical significance to people's daily travel planning and intelligent transportation system and is also the direction of our study. The traffic condition is a general concept that can be traffic speed, flow, or lane occupancy in our method. Without loss of generality, traffic flow was chosen as the traffic condition in our experiments.

Definition 1. Road network G: An undirected graph $G = (V, E, A)$ should be introduced in order to describe the relationship among adjacent nodes in the traffic network, where $V = \{v_i\}_{i=1}^{N}$ is a node set, and we treat each sensor distributed along the road as a node, N is the total number of nodes, and E is a set of edges which represents the connectivity among nodes in a transportation network. $A \in R^{N\times N}$ is a weighted adjacency matrix, and A_{v_i,v_j} represents the proximity from node v_i to node v_j.

Definition 2. Traffic flow: Traffic flow is defined as the number of vehicles passing on a certain road in a certain period of time. For example, given a time interval Δt, $y_{t+1}^{v_i}$ represents the traffic flow during the time horizon $(t, t + \Delta t)$ at the station v_i and t is the starting point.

Definition 3. Feature matrix $X^{N\times F\times S}$: The traffic flow of all nodes in the road network over S time slices are regarded as features, defined as $\mathrm{X} = (X_1, X_2, ..., X_S) \in R^{N\times F\times S}$, where N is the total number of nodes, S represents the number of the historical time slices, F is denoted as the feature of the historical time slices, which can be other traffic information, such as speed, traffic flow, lane occupancy rate. $X_t = (x_t^1, x_t^2, \ldots, x_t^N) \in R^{N\times F}$ represents all values of the features of N nodes at time of t , where x_t^i represents all the features of the node i at time of t.

Therefore, Short-term traffic flow prediction of all nodes can be defined as $Y_p = (y^1, y^2, \ldots, y^N) \in R^{N\times T_p}$, as shown by the blue part of Figure 2, where T_p is the length of predicting window. $y^i = (y_{t+1}^i, y_{t+2}^i, \ldots, y_{t+T_p}^i) \in R^{T_p}$ represents the traffic flow of the node i in time slice T_p, where y_t^i denotes the traffic flow of node i at time t. Generally speaking, traffic flow is a slow variable, so there is a strong correlation between the just past traffic data and the short-term future traffic data. The goal of our work is to establish a function $f(\cdot)$, which can extract the potential spatialtemporal relationship from the traffic flow in the past accurately, as shown in Figure 2.

It can be described by the following formula:

$$Y_p = f(X_s, G; \Theta), \tag{1}$$

where $X_s = (X_{t-T_s+1,} X_{t-T_s+2,}, \ldots, X_t) \in R^{N\times F\times T_s}$ is a historical time slice directly adjacent to the time slice to be predicted, G is a road network, and Θ denotes all learnable parameters in our model.

4. AGRGCN Method

4.1. Overview. Our task is mainly to extract meaningful patterns and features from historical traffic information, so as to predict the future traffic flow in different periods. Generally, short-term traffic prediction is of great significance for people's daily travel planning, urban road management, logistics supply chain, etc. Therefore, we follow the previous work [21, 22, 36–38] to study the short-term flow forecast based on GCN.

We propose a framework for spatial-temporal traffic flow prediction based on graph convolution and gated recurrent unit with attention mechanism. The framework is shown in Figure 3. It mainly contains 4 modules, data preprocessing, spatial feature extraction, temporal feature extraction, and results visualization. More details are as follows:

For the data processing module, the original traffic data will be normalized by min-max scaling $X_{\mathrm{norm}} = X - X_{\min}/X_{\max} - X_{\min}$ before feeding into the neural network. It can solve the problem of different scales of features in the traffic flow, which can also speed up gradient descent for optimal solutions for our model. The predefined adjacency matrix is constructed by the distance between each sensor on the traffic road network. The predefined adjacency matric $A \in R^{N\times N}$ is defined as follows:

$$A_{v_i,v_j} = \begin{cases} \dfrac{1}{\mathrm{dist}(v_i, \mathrm{v}_j)}, & \text{if } \mathrm{dist}(v_i, v_j) < \kappa, \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathrm{dist}(v_i, v_j)$ denotes the distance among notes v_i and v_j on the traffic road network, κ is the distance threshold among sensors. The distance among different nodes is measured by Google Maps.

The spatial feature extraction module consists of several graph convolution layers which can process sensor network graphs of non-Euclidean distances and extract the spatial dependencies among neighboring nodes in a traffic road network.

The temporal feature extraction module consists of several GRUs (gated recurrent units), which can capture potential temporal dependencies in traffic flow. Furthermore, an attention mechanism is introduced to adaptively evaluate the importance of past traffic flows at different time steps for future traffic flows.

In the results visualization module, the output values of the model are denormalized and then plotted as curves and compared with the curves of the original values.

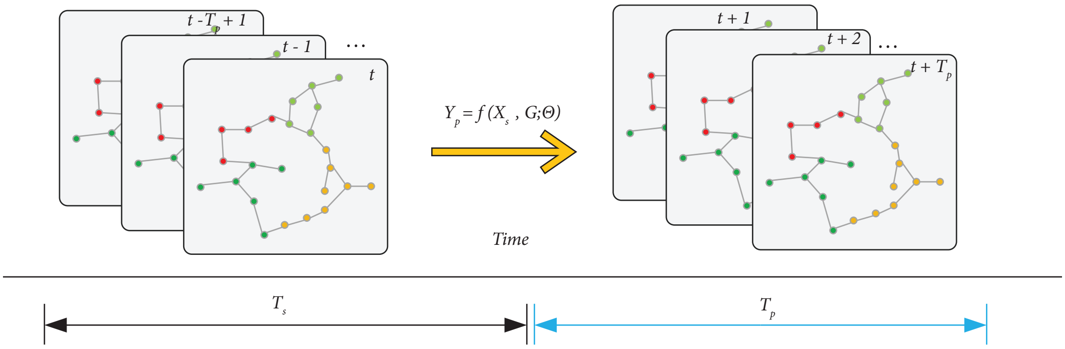


FIGURE 2: The goal of traffic forecasting is to fit a non-linear function that extracts the spatial and temporal dependencies in past traffic flows, and thus, accurately predicts future traffic flows.

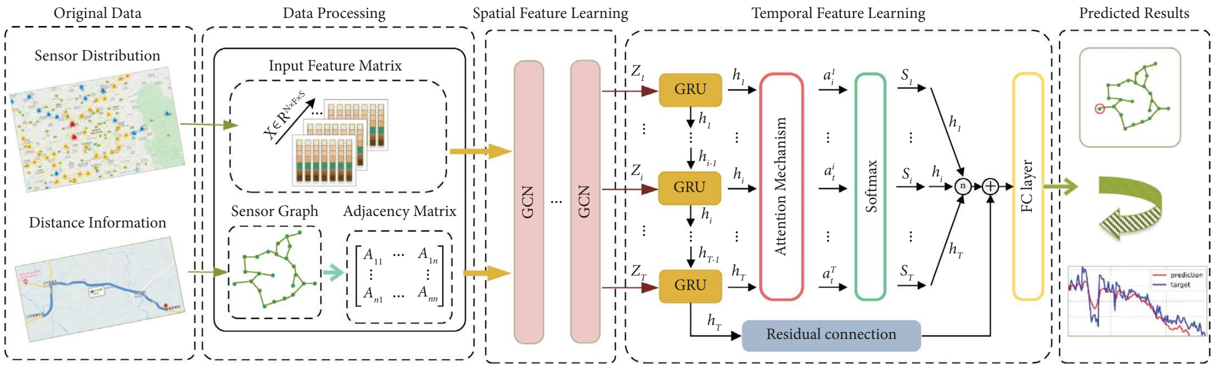


FIGURE 3: The overall architecture of the AGRGCN framework.

4.2. Extracting Spatial Dependencies. Spatial dependencies are a crucial factor in traffic flow prediction. There is a strong spatial correlation among different nodes in a traffic network. The nature of a traffic network is a graph structure, and each node can be considered as a signal on the graph. Therefore, a spectral graph-based approach is used to capture patterns and features in a spatial sense to take full advantage of the topological properties of traffic networks. This approach extends the convolution operation to the domain of graph-structured data, which treats the data as signals on a graph, and then processes it directly on the graph.

In spectral graph analysis, the graph is represented by the corresponding Laplacian matrix $L = D - A$, where $A \in R^{N\times N}$ is an adjacency matrix. $D \in R^{N\times N}$ is a degree matrix, which is a diagonal matrix, $D_i = \sum_j^n A_{ij}$ represents the degree of node i in a graph. In most recent work on traffic flow prediction, a symmetric normalized Laplacian matrix L_{sym} is used to describe spatial relationships among nodes in a traffic network.

$$L_{\text{sym}} = I_N - D^{-(1/2)} A D^{-(1/2)}, \tag{3}$$

where I_N is an identity matrix.

According to [22], graph convolution operation can be approximated by expansion of first-order Chebyshev polynomial, as shown in the following formula:

$$H^{(l+1)} = \sigma\left(L_{\text{sym}} H^{(l)} W^{(l)}\right), \tag{4}$$

where $H^{(l)} \in R^{N*D}$ and $H^{(0)} = X \in R^{N\times C}$ denote the output and input of the GCN layer l, C is the input dimension and D is the hidden dimension, and $\sigma(\cdot)$ denotes the activation function.

In our research, a two-layer graph convolutional network is used as a spatial feature extraction component in our model, which can be formulated as follows:

$$Z = f(X, A) = \text{ReLu}\left(L_{\text{sym}} \text{ReLU}\left(L_{\text{sym}} X W^{(0)}\right) W^{(1)}\right), \tag{5}$$

where $Z \in R^{N\times D}$ denotes the output and $\text{ReLU}(\cdot)$ is used as the activation of the network.

4.3. Extracting Temporal Dependencies. Except for the spatial dependencies, the complex temporal dependencies are also involved in traffic flow prediction. RNN is widely used to deal with time-series data. However, the classical RNN has some limitations for long-term prediction due to the problems of gradient disappearance and gradient explosion. LSTM and GRU are both variants of RNN and can solve the above problem of RNN by gating units. At the same time, it can have a better memory effect on the time series data of long and short time. Compared with LSTM, GRU has a relatively lightweight model, fewer parameters, and faster training ability [35]. Therefore, GRU is selected in our research to extract the time dependencies of traffic flow. The spatial module from the previous section is selected to replace the MLP part of the GRU so that the spatialtemporal dependencies in the nodes can be extracted simultaneously. It can be defined by the formula as follow:

$$\begin{aligned} u_t &= \mathrm{Sigmoid}\left(W_u\left[f\left(A, X_t\right), h_{t-1}\right] + b_u\right), \\ r_t &= \mathrm{Sigmoid}\left(W_r\left[f\left(A, X_t\right), h_{t-1}\right] + b_r\right), \\ \widetilde{h}_t &= \tanh\Big(W_{\widetilde{h}}\left[f\left(A, X_t\right), \left(r_t \odot h_{t-1}\right)\right] + b_{\widetilde{h}}\Big), \\ h_t &= u_t \odot h_{t-1} + \left(1 - u_t\right) \odot \widetilde{h}_t, \end{aligned} \tag{6}$$

where $f\left(A, X_t\right)$ and h_t denote input and output at time step t, $[\cdot]$ represents the concatenate operation, $\odot$ stands for the Hadamard product of two matric, and u and r stand for update gate and reset gate, respectively. W_r, W_u, $W_{\widetilde{h}}$, b_r, b_u, and $b_{\widetilde{h}}$ are learnable parameters.

To predict the future traffic flow more accurately, GRU-based network modules are used to mine the temporal features in the traffic flow. However, according to Cho et al. [10], the prediction performance of such RNN-based networks deteriorates rapidly as the length of the input sequence increases. To alleviate this problem, an attention mechanism is integrated into the temporal feature extraction module to achieve more accurate future traffic flow prediction. This attention mechanism enables adaptive evaluation of the impact of hidden states on future traffic flow at different time steps. More specifically, the attention weights of the hidden states at each time step are calculated by the following equation:

$$\begin{aligned} u_t^i &= U_w \tanh\left(W_w h_t^i + b_w\right), \\ \alpha_t^i &= \frac{\exp\left(u_t^i\right)}{\sum_{j=1}^{T} \exp\left(u_t^i\right)}, \end{aligned} \tag{7}$$

where W_w, U_w, and b_w are learnable parameters and the attention weight α_t^i represents the importance of different time steps on traffic flow prediction in the past. In order to integrate the impact of traffic flow at different time steps on future traffic flow, the weighted sum of the hidden states at different time steps is obtained by the attention mechanism to acquire the context vector S_t.

$$S_t = \sum_{i=1}^{T} a_{it} h_{it}. \tag{8}$$

In addition, a jump connection is introduced in the temporal feature extraction module to alleviate the model degradation problem. The last hidden state of GRU is jump connected after the attention mechanism module, as shown in Figure 3. The specific formulation is as follows:

$$o_t = \mathrm{ReLU}\left(S_t + h_t^{\mathrm{last}}\right). \tag{9}$$

Finally, the prediction sequence of traffic flow $Y_p = \left(y^1, y^2, \ldots, y^N\right) \in R^{N \times T_p}$ is gained by a fully connected layer.

$$Y_p = \mathrm{ReLU}\left(w_p' o_t + b_p'\right), \tag{10}$$

where w_p', b_p' denote the weight and bias of the fully connected layer, respectively.

5. Experiments

5.1. Datasets. To evaluate the performance and robustness of our model, extensive experiments are conducted on two different time-granularity datasets.

> PeMSD4: It is collected by the Caltrans Performance Measurement System (PeMS) with a time granularity of 5 minutes. The PeMSD4 dataset refers to the traffic flow data in the San Francisco Bay Area, which contains traffic information of 307 loop detectors from 1/Jan/2018 to 28/Feb/2018.
>
> HW-ENG: This traffic dataset contains specific traffic information, including average speed, traffic flow, sensor location, and date, collected from 222 detectors on the highway of England. The time granularity of the original traffic dataset is 15 minutes. Sensors are distributed on 12 roads, including M6, M60, M62, M67, and A556, which cover server cites that contain Manchester, Warrington, and Blackburn. The distribution of sensors of the dataset is presented in Figure 4. A whole year of traffic data ranging from January 1st, 2019, to December 31st, 2019, is used for the experiment. The total number of traffic data in the dataset is 15,472,512. To mine for hidden correlations among traffic flows at nearby observation points, we enhanced the dataset with topological road information based on the distance among sensors through Google Maps' service.

In order to study the traffic conditions of the two datasets so as to analyze the subsequent experimental results more rationally, some statistical analyses are performed; the results are shown in Figure 5. More specifically, Figure 5(a) and 5(b), respectively, show the distribution of speed data and traffic flow data in the two datasets. The speed distribution of PeMSD4 is relatively simple, with more than 80% of data in the range of 60–65 mile/h, indicating that the traffic condition is relatively simple. The distribution of speed and traffic flow of HW-ENG is more uniform than that of PeMSD4.

To analyze the spatial correlation among different nodes, the Pearson correlations among traffic flows at different measurement points in the traffic network are calculated, and their distributions are statistically presented, in Figure 5(c). Pearson's correlation coefficient $\rho_{X,Y}$ is used to measure the degree of linear correlation between variables X and Y of two data sets. The closer the absolute value is to 1, the stronger the correlation is, conversely, the closer it is to 0,

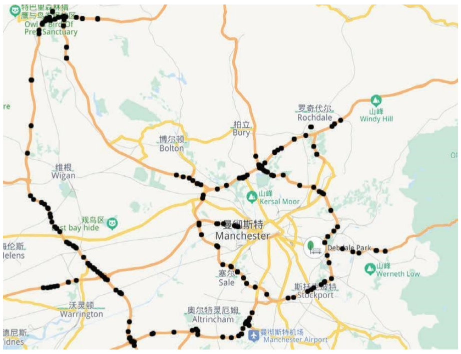
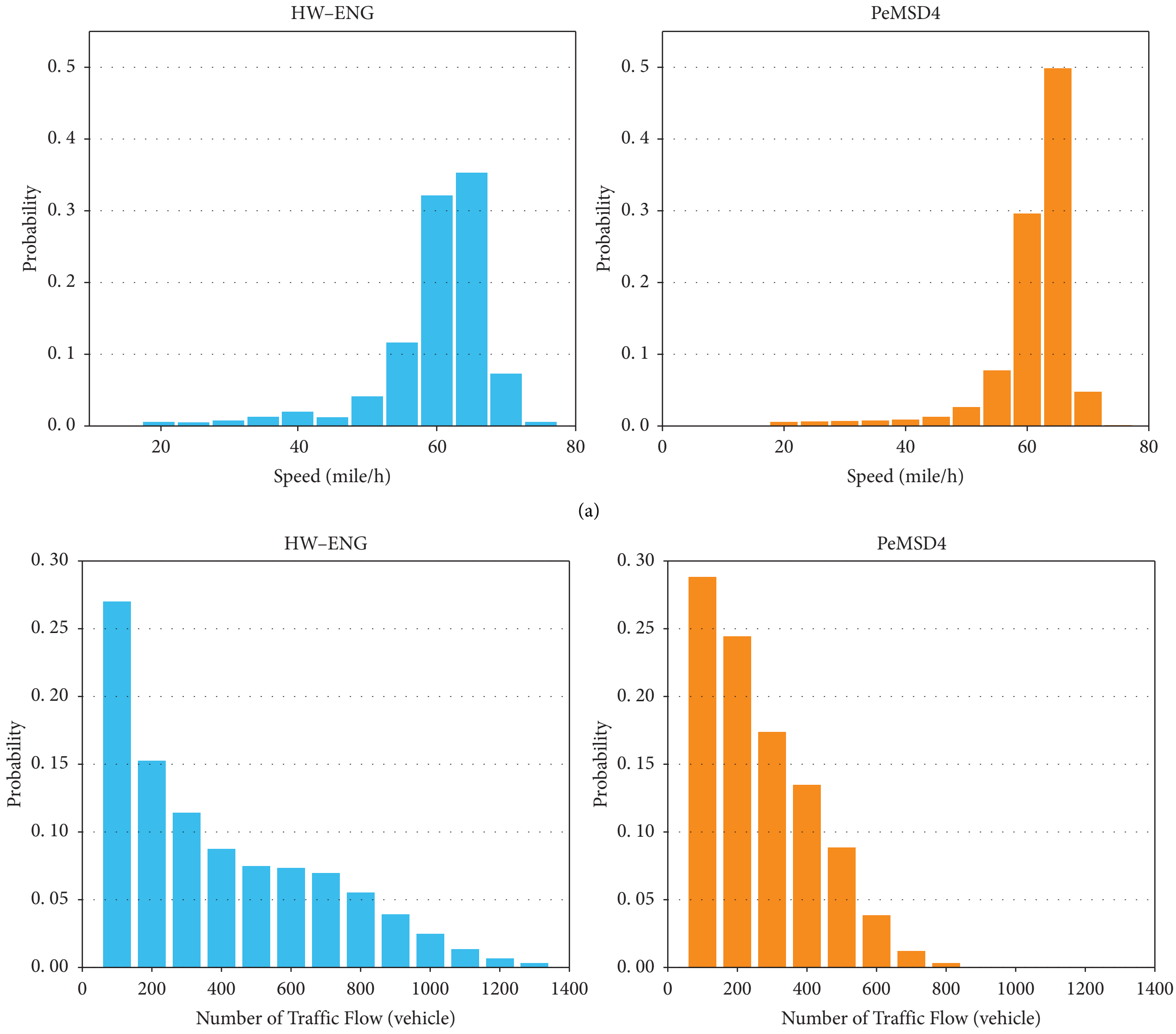


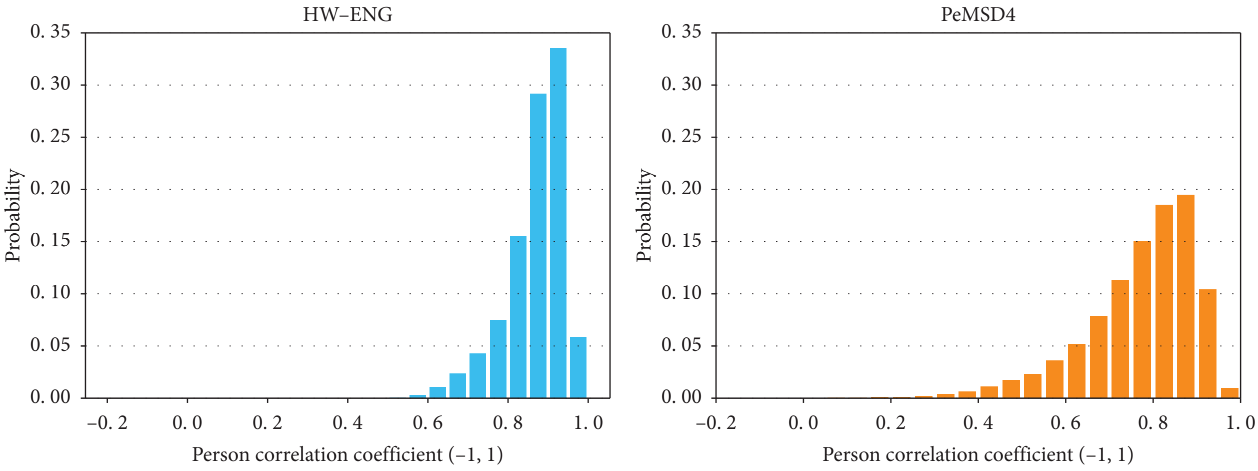
Figure 4: The distribution of monitoring points in the HW-ENG dataset.



(a)

(b)

Figure 5: Continued.



(c)

Figure 5: Distribution of traffic information in the PEMSD4 and HW-ENG datasets: (a) distribution of speed data; (b) distribution of traffic flow data; (c) distribution of correlations between nodes in the traffic network.

the weaker the correlation. The specific formula for $\rho_{X,Y}$ is as follows:

$$\begin{aligned}\rho_{X,Y} &= \frac{\operatorname{cov}(X,Y)}{\sigma_X\sigma_Y}\\ &= \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y} - 1 < \rho_{X,Y} < 1.\end{aligned} \tag{11}$$

It can be obtained from Figure 5(c), there is a strong spatial correlation among nodes in the HW-ENG dataset, of which 84% are greater than 0.8. The correlation among nodes in the PeMSD4 dataset is relatively low, and there are some pairs of nodes with low correlation. Therefore, compared with PeMSD4, the traffic condition of HW-ENG is more complex and the correlation among nodes is stronger.

A more detailed comparison of these two datasets is shown in Table 1. The total number of edges among nodes in PeMSD4 is less than HW-ENG, which indicates that the topological information in the traffic flow is insufficient to be described in PeMSD4. The key to improving the prediction performance is effectively capturing the spatial dependencies among nodes.

5.2. Setting

5.2.1. Baselines. AGRGCN and some traditional statistical-based and deep learning-based methods are compared for traffic flow prediction, and these methods can reflect the recent progress in this field. The baselines are introduced as follows:

HA (historical average) method: The average value of the last 8 steps is used to predict the next value.

ARIMA (autoregressive integrated moving average model): It fits a parametric model with the data observed in the past week and then predicts the future traffic flow.

SVR (Support vector machine) for Regression: A regression method with good generalization ability is widely used for the prediction of time-series data.

MLP (multilayer perceptron): (One input layer, one or more hidden layers, and one output layer), which can solve non-linear problems by using activation functions to simulate neurons. The number of hidden layer cells is set to 64.

GRU (gated recurrent unit) network: A variant of the RNN model which powerful in capturing sequential dependencies.

T-GCN [35]: It combines graph convolution network (GCN) and gated recurrent unit (GRU) and can capture time and space dependencies.

HGCN [21]: It is a combination of the graph convolutional network (GCN) and feedforward neural network (FNN).

ASTGCN [17]: It consists of several graph convolution components with a spatialtemporal attention mechanism.

AGCRN [38]: It can construct a road traffic network diagrams adaptively without the need for pre-defined adjacency matrices.

5.2.2. Experimental Setup. The original traffic data is processed on a PC (CPU: AMD Ryzen 4600H with Radeon Graphics @ 3.0 GHz, GPU: NVIDIA GeForce GTX 1650 with 4 GB of GPU memory, memory: 16 GB). Moreover, with the support of the Python libraries that include NumPy and PyTorch. The deep learning model is implemented on a Dell R730 server (CPU: E5-2603V4 ∗ 2 @ 1.2–3.2 GHz, GPU: Nvidia Tesla K80 with 12 GB of GPU memory, memory: 128 GB). The architecture of our model framework consists of two graph convolutional layers, one gated recurrent layer, one attention layer, one residual connection

Table 1: Comparison of properties of the two datasets.

Dataset	Samples	Nodes	Edges	Time range	Time granularity (min)	Average flow	Average speed (mile/h)
PeMSD4	10,433,088	307	340	2018/1/1~2018/2/28	5	211.80	63.47
HW-ENG	15,472,512	222	1886	2019/1/1–2019/12/31	15	340.57	61.06

layer, and one fully connected layer. For the HW-ENG dataset, the first 10 months are used for training data, and the last 2 months are used for test data. For PeMSD4, the first 45 days were used as the training data, and the last 15 days were used as the test date. Besides, the hyper-parameters settings of the model are as follows: the mini-batch size is set to 128. The dropout rate is set to 20%. An Adam optimizer is used to update the model parameters with an initial learning rate of 0.001. The mean square error L_f is chosen as the loss function with the following equation:

$$L_f = \frac{1}{n} \sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2, \tag{12}$$

where y_i denotes the ground truth and $\hat{y}_i$ denotes the predicted values.

5.2.3. Evaluation Metric. The performance of all models is evaluated by the following three metrics: mean absolute error (MAE), mean absolute percentage error (MAPE), and rooted mean square error (RMSE), which are the most widely used methods in traffic flow prediction problems.

$$\begin{aligned} \text{MAE} &= \frac{1}{N} \sum_{i=1}^{N} \left|y_i - \hat{y}_i\right|, \\ \text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i}^{n} \left(y_i - \hat{y}_i\right)^2}, \\ \text{MAPE} &= \frac{100\%}{N} \sum_{i=1}^{N} \left|\frac{y_i - \hat{y}_i}{y_i}\right|, \end{aligned} \tag{13}$$

where y_i represents the ground truth while $\hat{y}_i$ denotes the predicted values and N denotes the observed samples.

5.3. Results and Analysis

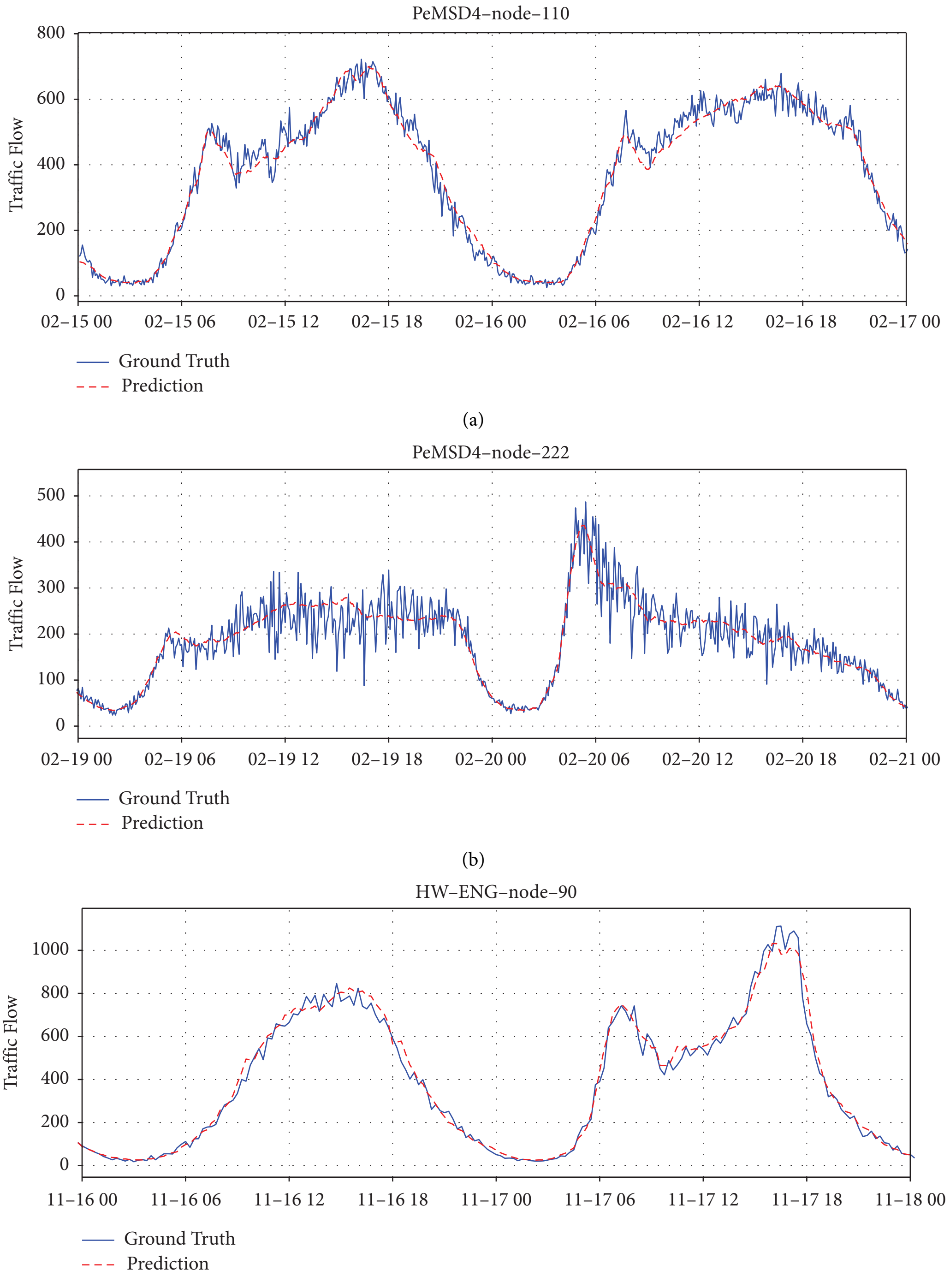
5.3.1. Overall Result. To verify the accuracy of the AGRGCN prediction results, the predicted values are compared with the original values. As shown in Figure 6, the traffic flow curves of some randomly selected nodes on two different time-granularity datasets are plotted, where the blue line represents the true value and the red line indicates the predicted value. Obviously, the trends and fluctuations of the two curves are highly consistent, which proves the effectiveness and robustness of the performance of our model AGRGCN.

Tables 2 and 3 show the quantitative results of these models for 15-, 30-, and 60-minute traffic flow prediction, including MAE, RMSE, and MAPE. AGRGCN achieves better prediction performance on both datasets, especially on multistep prediction (3%-4% improvement on PeMSD4). In addition, the prediction performance advantage of our model is pronounced under more complex traffic conditions (9%–16% improvement on HW-ENG). In addition, our method has a lower computational cost relative to AGCRN [38] and ASTGCN [17], with a reduction of 53.6% on the PeMSD4 dataset and 88.6% on the HW-ENG dataset. Considering the improved performance and lower computational cost of our model, our model outperforms the baselines.

Traditional methods, such as HA, ARIMA, and SVR, have a strong dependency on the processing of original input data and the prediction performance is much worse than that of deep learning-based methods. The prediction performance of MLP and GRU is better than traditional methods, but these methods have some limitations because they are not suitable for handling non-Euclidean distance structure data, so only the time dependencies of traffic flow can be captured, while the spatial dependencies is ignored.

Based on the graph convolution method, HGCN [21], T-GCN [35], ASTGCN [17], and AGCRN [38] can capture the spatialtemporal dependencies in traffic flow simultaneously and the prediction performance is better than MLP and GRU. The performance of HGCN [21] and T-GCN [35] is weaker than that of ASTGCN [17] and AGCRN [38] due to the limited representation capability of the model. ASTGCN [17] adopts the spatialtemporal attention mechanism module, which can better extract the spatialtemporal dependencies in traffic flow, but the multistep prediction performance is lower than that of AGCRN [38]. AGCRN [38] extracts the spatial dependencies of traffic flow by adaptively constructing the adjacency matrix and extracts the temporal dependencies of traffic flow by the GRU module. In both datasets, the prediction performance of AGCRN [38] is the best among the baselines. Compared with AGCRN [38], AGRGCN achieves better prediction results on most prediction time steps on PeMSD4 and HW-ENG datasets, except for certain metrics predicted in the short-term range (e.g. 15 min). Especially on HW-ENG with a larger time-granularity and more complex road conditions, the performance of AGRGCN is superior to AGCRN [38], which illustrates the adaptability of AGCRN [38] and its effectiveness in capturing the spatialtemporal dependencies in traffic flow.

Figure 7 shows the comparison of the prediction performance of each method at different time steps on the PeMSD4 and HW-ENG datasets. On PeMSD4, AGCRN [38] performs best on short time steps (15 min). This may be due to the inadequate description of road topology information in the PeMSD4 dataset (as known from the statistical analysis of the dataset in the previous section). As a result, models constructed based on this adjacency matrix (e.g., ASTGCN [17], and AGRGCN) have difficulty adequately

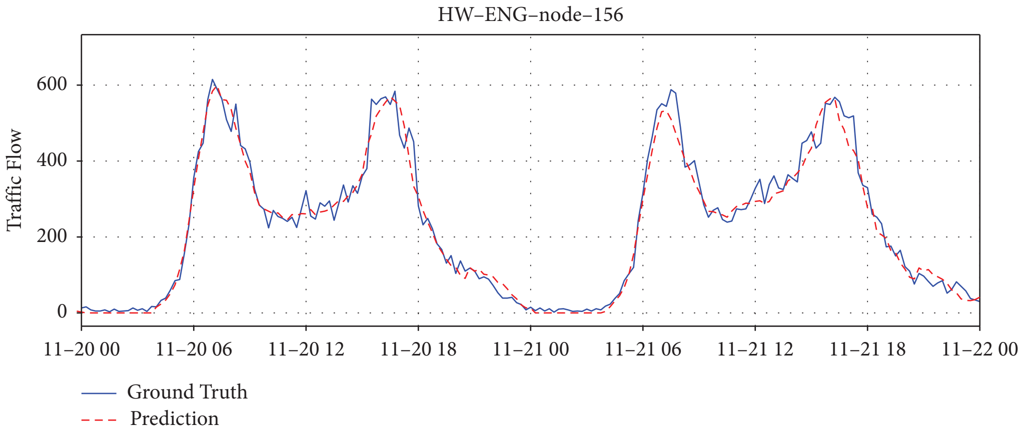


(a)

(b)

(c)

Figure 6: Continued.



(d)

Figure 6: Comparison between the real traffic flow and the predicted values for some randomly selected nodes on two different time-granularity datasets.

Table 2: The prediction performance of different model on the PeMSD4 dataset.

Methods	15 min			45 min			1 h			Training time (s/epoch)
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	
HA	28.61	42.76	20.00	39.14	57.40	28.4	46.39	67.38	34.60	—
SVR	26.44	41.96	18.50	37.66	57.62	25.7	43.46	66.49	28.90	—
ARIMA	21.8	34.05	14.10	32.91	49.76	21.8	38.97	58.24	26.50	—
MLP	21.47	33.2	14.40	28.67	42.75	20.6	32.77	47.88	25.40	5
GRU	21.63	33.97	15.70	24.11	37.05	17.8	24.56	37.74	18.30	26
T-GCN	22.15	33.88	17.30	28.27	41.99	23.1	31.34	46.47	24.50	30
HGCN	21.1	33.28	14.60	27.81	42.45	19.6	31.45	47.2	23.00	12
ASTGCN	19.9	31.15	14.00	24.26	37.12	17.9	26.8	40.47	20.90	150
AGCRN	**19.22**	**30.08**	**13.20**	21.43	**33.29**	14.5	22.25	**34.42**	15.40	110
AGRGCN	20.01	33.03	13.80	**21.04**	34.65	**14.4**	**21.52**	35.28	**14.90**	51

The bold values indicate the best performance metrics in the performance comparision.

Table 3: The prediction performance of different model on the HW-ENG dataset.

Methods	15 min			45 min			1 h			Training time (s/epoch)
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	
HA	76.94	115.62	39.80	95.03	142.18	49.9	104.96	156.04	56.10	—
SVR	57.06	102.21	27.70	78.38	133.03	38.4	89.22	149.64	43.40	—
ARIMA	35.5	57.34	17.70	70.57	115.26	31.6	88.73	142.65	40.00	—
MLP	30.29	48.51	15.10	49.62	77.80	27.3	59.01	90.58	34.00	4
GRU	29.7	47.46	15.20	43.08	68.86	21.0	50.79	78.75	28.60	12
T-GCN	30.47	48.42	16.00	49.46	77.54	26.6	54.43	83.71	29.70	33
HGCN	29.23	46.97	15.30	46.56	72.93	23.8	53.43	84.21	25.60	11
ASTGCN	26.98	43.25	15.70	45.70	73.08	21.8	39.57	60.03	22.80	120
AGCRN	28.72	45.23	17.80	35.64	57.57	19.5	35.92	58.81	19.40	105
AGRGCN	**26.06**	**42.23**	**14.80**	**33.12**	**54.27**	**17.5**	**31.16**	**52.2**	**16.30**	12

The bold values indicate the best performance metrics in the performance comparision.

capturing traffic flow's spatial dependencies. In contrast, AGCRN [38] benefits from the learnable adjacency matrix, and thus, achieves better prediction performance at short time steps. Compared to the baselines, AGRGCN benefits from the ability to adaptively extract dependencies among traffic flows at different time steps at long time steps. What is more, the prediction performance of AGRGCN deteriorates much more slowly than other models, which illustrates the robustness of our model.

Compared with AGCRN [38], AGRGCN achieves better prediction results on most prediction time steps for the PeMSD4 and HW-ENG datasets, except for certain metrics

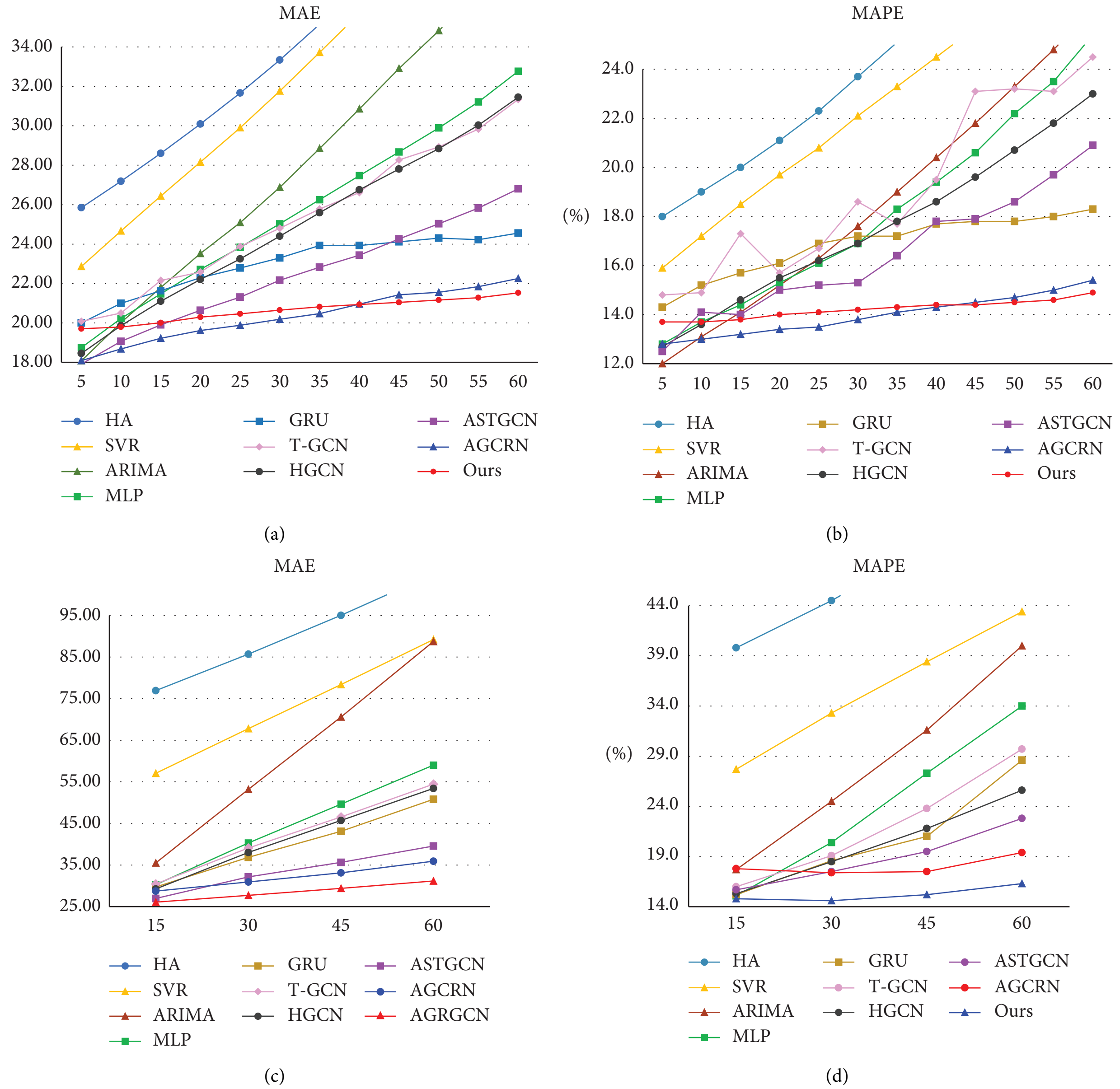


FIGURE 7: Prediction performance comparison at each time step on the PeMSD4 and HW-ENG datasets.

(e.g., 15 and 30 minutes) predicted in the short-term range. In particular, on HW-ENG, the limitations of the baselines are more obvious due to the larger time granularity and more complex road conditions, which illustrates the adaptability and robustness of AGRGCN on different types of traffic datasets.

5.3.2. Ablation Study. In this part, an ablation study was conducted to validate the impact of different modules in our model on the HW-ENG dataset. GCN was chosen as the benchmark comparison method, which contains 2 layers of graph convolution layers. Meanwhile, three variants of the AGRGCN model have been designed as follows:

AGCN: the GRU module in AGRGCN is replaced with a fully connected layer

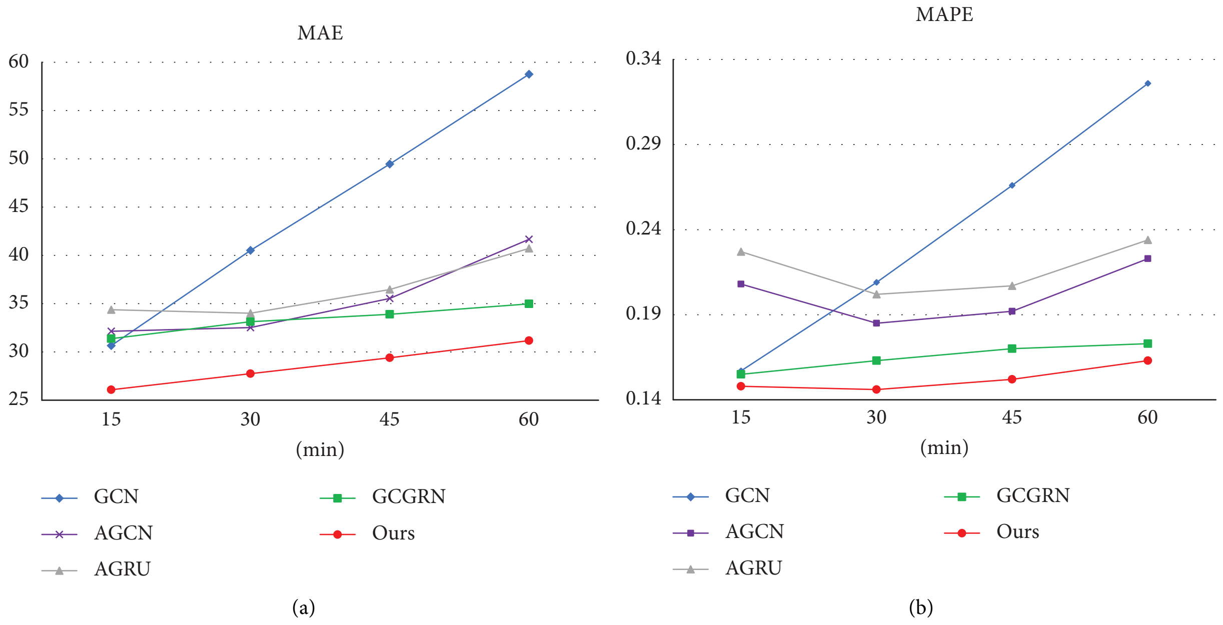
AGRN: the graph convolution module in AGRGCN is removed

GCGRN: the attention mechanism in AGRGCN is removed

As can be seen from the results in the Figure 8

(1) Without the GRU module, it is difficult for AGCN to capture temporal dependencies, and although there is good performance for a short time, the model performance deteriorates rapidly in multistep prediction

(2) Without the GCN module, AGRU is unable to capture the spatial dependencies of traffic flow, and the overall performance is poor despite the memory capability brought by the GRU module of the model



(a)

(b)

Figure 8: Ablation experiments performed on the HW-ENG dataset.

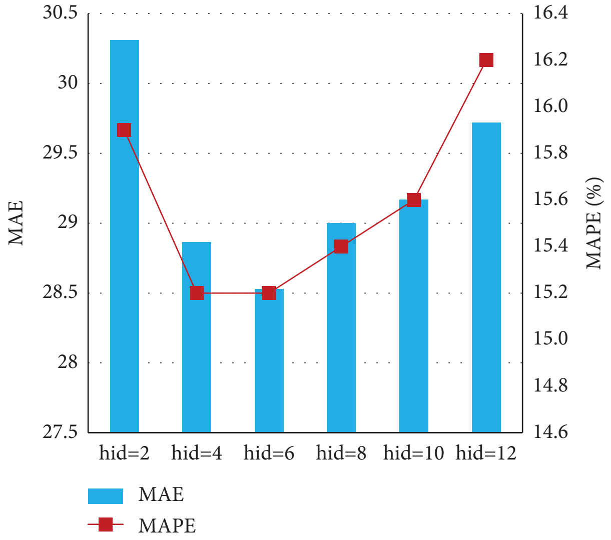


Figure 9: Effect of different GRU hidden units for the performance of AGRGCN on the HW-ENG dataset.

(3) GCGRU can capture spatialtemporal dependencies, but the model performance decreases compared with AGRGCN due to the lack of attention module

(4) AGRGCN achieved the best performance, illustrating that the attention mechanism's introduction can improve the model's performance.

Overall, this ablation experiment demonstrates the ability of AGRGCN to capture the temporal and spatial dependencies in the traffic flow. At the same time, the indispensable role of each submodule of our model is validated.

5.3.3. *Model Analysis.* One key parameter in our model is the number of GRU hidden units of each node, which not only affects the overall number of parameters of our model but also the learning ability of AGRGCN to capture the spatialtemporal dependencies. Figure 9 shows the influence of different GRU hidden units on the performance of AGRGCN on the HW-ENG dataset. When the number of GRU hidden units is set to 6, the performance of the model achieves the best. Besides, either the number of the hidden layer is set larger or smaller would degrade model performance. On the one hand, the model representation capability is limited when the number is small. On the other hand, a large number of GRU hidden units lead to an increase of learnable parameters, making model training more complex.

6. Conclusion

In this paper, an attention-based spatialtemporal graph convolution framework, named AGRGCN, is proposed and used for traffic flow prediction. This model combines GCN and recurrent neural networks to extract the spatial-temporal dependencies in traffic flows efficiently. In addition, an attention mechanism is introduced into our model to learn the relationship among different time steps adaptively.

Extensive experiments with 10 models on two different time-granularity datasets demonstrate the validity and robustness of our models and their various submodules. Each model is evaluated in terms of its prediction accuracy and training time cost. Considering the improved performance and lower computational cost of our model, our model is better than the baselines and the advantage is more obvious for a longer time step. In the future, we will expand our work by focusing on the following aspects:

(1) To achieve more accurate traffic flow prediction, we will introduce some external factors (weather, holidays, etc.) into our model

(2) Due to the strong daily and weekly periodicity of traffic flow, data from the same time period of past days and weeks will be used as input features to predict traffic flows for the same time period of future days

Data Availability

The HW-ENG dataset is derived from data published on the official Highways UK website (https://tris.highwaysengland.co.uk/), the authors enhanced it and released it to the open source community (https://github.com/WhutWzh/HW-ENG-dataSet) under a government open source li-cense (https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/). The PEMSd4 dataset is an open source dataset that can be downloaded from the developer community (https://github.com/Davidham3/ASTGCN).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

P. L. and Z. H. performed conceptualization, methodology, and investigation and prepared the original draft; P. L., Z. W., and J. H. performed validation; Q. L. and Q. W. performed formal analysis;; J. H., Q. L., and Q. W. wrote, reviewed, and edited the manuscript; Z. W. performed visualization; P. L. and J. H. supervised the study; P. L. and J. H. performed project administration; P. L., J. H., and Q. L. acquired funding. All authors have read and agreed to the published version of the manuscript. Authorship must be limited to those who have contributed substantially to the work reported.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (NSFC), grant number 52075404, the National Key Research and Development Project of China, grant number 2020YFB1710804, and the Application Basic Frontier Special Project of Wuhan Science and Technology Bureau, grant number 2020010601012176.

References

[1] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] X. Chen, S. Wu, C. Shi et al., "Sensing data supported traffic flow prediction via denoising schemes and ann: a comparison," *IEEE Sensors Journal*, vol. 20, no. 23, pp. 14317–14328, 2020.

[3] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.

[4] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, pp. 1–11, 1984.

[5] E. Siggiridou and D. Kugiumtzis, "Granger causality in multivariate time series using a time-ordered restricted vector autoregressive model," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1759–1773, 2016.

[6] X. Zhang, X. Zheng, F. Jiang, Z. G. Zhang, M. Katakowski, and M. Chopp, "Dual-color fluorescence imaging in a nude mouse orthotopic glioma model," *Journal of Neuroscience Methods*, vol. 181, no. 2, pp. 178–185, 2009.

[7] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

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

[9] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Wuhan, China, November 2016.

[10] K. Cho, B. van Merrienboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[11] H. Miao, Y. Zhang, D. Wang, and S. Feng, "Multi-output learning based on multimodal GCN and Co-attention for image aesthetics and emotion analysis," *Mathematics*, vol. 9, no. 12, p. 1437, 2021.

[12] S. Du, T. Li, X. Gong, and S. J. Horng, "A hybrid method for traffic flow forecasting using multimodal deep learning," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 85–97, 2020.

[13] P. Sun, A. Boukerche, and Y. Tao, "SSGRU: a novel hybrid stacked GRU-based traffic volume prediction approach in

a road network," *Computer Communications*, vol. 160, pp. 502–511, 2020.

[14] T. Sun, S. Zhu, R. Hao, B. Sun, and J. Xie, "Traffic missing data imputation: a selective overview of temporal theories and algorithms," *Mathematics*, vol. 10, no. 14, p. 2544, 2022.

[15] J. Long, L. Liu, H. Fei et al., "Contextual semantic-guided entity-centric GCN for relation extraction," *Mathematics*, vol. 10, no. 8, p. 1344, 2022.

[16] J. Guo, C. Zheng, S. Li, Y. Jia, and B. Liu, "BiInfGCN: bilateral information augmentation of graph convolutional networks for recommendation," *Mathematics*, vol. 10, no. 17, p. 3042, 2022.

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

[18] Y. Li, W. Zhao, and H. Fan, "A spatio-temporal graph neural network approach for traffic flow prediction," *Mathematics*, vol. 10, p. 1754, 2022.

[19] S. Yang, H. Li, Y. Luo, J. Li, Y. Song, and T. Zhou, "Spatiotemporal adaptive fusion graph network for short-term traffic flow forecasting," *Mathematics*, vol. 10, no. 9, p. 1594, 2022.

[20] X. Chen, J. Ling, S. Wang, Y. Yang, L. Luo, and Y. Yan, "Ship detection from coastal surveillance videos via an ensemble canny-Gaussian-morphology framework," *Journal of Navigation*, vol. 74, no. 6, pp. 1252–1266, 2021.

[21] T. Zhang, W. Ding, T. Chen, Z. Wang, and J. Chen, "A graph convolutional method for traffic flow prediction in highway network," *Wireless Communications and Mobile Computing*, vol. 2021, p. 8, Article ID 1997212, 2021.

[22] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting 2018," 2017, https://arxiv.org/abs/1707.01926.

[23] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, pp. 249–254, 1995.

[24] 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, pp. 664–672, 2003.

[25] D. D. Chuwang and W. Chen, "Forecasting daily and weekly passenger demand for urban rail transit stations based on a time series model approach," *Forecasting*, vol. 4, pp. 904–924, 2022.

[26] C. H. Wu, J. M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[27] M.-W. Li, W.-C. Hong, and H.-G. Kang, "Urban traffic flow forecasting using gauss–SVR with cat mapping, cloud model and PSO hybrid algorithm," *Neurocomputing*, vol. 99, pp. 230–240, 2013.

[28] C. H. Yang, J. C. Shao, Y. H. Liu, P. H. Jou, and Y. D. Lin, "Application of fuzzy-based support vector regression to forecast of international airport freight volumes," *Mathematics*, vol. 10, no. 14, p. 2399, 2022.

[29] X. Wang, Y. Ma, S. Huang, and Y. Xu, "Data imputation for detected traffic volume of freeway using regression of multilayer perceptron," *Journal of Advanced Transportation*, vol. 2022, Article ID 4840021, 16 pages, 2022.

[30] A. Crivellari and E. Beinat, "Forecasting spatially-distributed urban traffic volumes via multi-target LSTM-based neural network regressor," *Mathematics*, vol. 8, no. 12, p. 2233, 2020.

[31] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[32] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[33] W. Jin, Y. Lin, Z. Wu, and H. Wan, "Spatio-temporal recurrent convolutional networks for citywide short-term crowd flows prediction," in *Proceedings of the Proceedings of the 2nd International Conference on Compute and Data Analysis - ICCDA 2018*, pp. 28–35, DeKalb, IL, USA, March 2018.

[34] L. Bai, L. Yao, S. S. Kanhere, Z. Yang, J. Chu, and X. Wang, "Passenger demand forecasting with multi-task convolutional recurrent neural networks," in *Proceedings of the Advances in Knowledge Discovery and Data Mining*, Q. Yang, Z.-H. Zhou, Z. Gong, M.-L. Zhang, and S.-J. Huang, Eds., pp. 29–42, Springer International Publishing, Berlin, Germany, 2019.

[35] 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.

[36] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3634–3640, Stockholm, Sweden, July 2018.

[37] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 890–897, 2019.

[38] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," 2020, https://arxiv.org/abs/2007.02842.