Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 7682274, 17 pages
https://doi.org/10.1155/2022/7682274

Research Article

An Attention Encoder-Decoder Dual Graph Convolutional Network with Time Series Correlation for Multi-Step Traffic Flow Prediction

Shanchun Zhao and Xu Li

School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China

Correspondence should be addressed to Shanchun Zhao; zhao_sc_2021@163.com

Received 21 January 2022; Revised 2 March 2022; Accepted 17 March 2022; Published 9 April 2022

Academic Editor: Yajie Zou

Copyright © 2022 Shanchun Zhao and Xu Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Accurate traffic prediction is a powerful factor of intelligent transportation systems to make assisted decisions. However, existing methods are deficient in modeling long series spatio-temporal characteristics. Due to the complex and nonlinear nature of traffic flow time series, traditional methods of prediction tasks tend to ignore the heterogeneity and long series dependencies of spatio-temporal data. In this paper, we propose an attentional encoder-decoder dual graph convolution model with time-series correlation (AED-DGCN-TSC) for solving the spatio-temporal sequence prediction problem in the traffic domain. First, the time-series correlation module calculates the sequence similarity by fast Fourier transform and inverse fast Fourier transform, while obtaining multiple possible lengths as possible solutions for the sequence period length. Then, K possible periods fetches are selected and the corresponding sequences are weighted and aggregated to the target sequence. Then, the gated dual graph convolution recurrent unit uses the graph convolution operation, which combines the ideas of node embedding, and dual graph, as an operation inside the gated recurrent structure to capture the spatio-temporal heterogeneity relationship of long sequences. The gated decomposition recurrent module decomposes the time series into the period and trend terms, which are modelled by convolutional gated recurrent unit (ConvGRU) and then fused with features, respectively, and output after graph convolution. Finally, multi-step prediction of future traffic flow is performed in the form of encoder-decoder. Experimental evaluations are conducted on two real traffic datasets, and the results demonstrate the effectiveness of the proposed model.

1. Introduction

Transportation affects people's daily travel and plays an important role in our lives. With the development of urban construction, the population also grows with it, while posing challenges to urban planning. Traffic forecasting is an important part of the intelligent transportation system and also an essential tool for traffic decision and guidance. Traffic forecasting is the process of predicting future road traffic conditions by compiling historical data as a reference and using specific methods. Effective prediction of traffic flow characteristics can improve the efficiency of the road network and relieve the pressure on the road network.

Traditional research methods have mostly dealt with the problem of predicting future traffic conditions linearly through time-series-related methods. Typical methods of such classical statistics include Markov chain [1], autoregressive integrated moving average (ARIMA) [2], linear regression [3], and fuzzy time-series techniques [4]. Unfortunately, traditional statistical methods show limitations in solving such a nonlinear prediction problem of traffic flow, and the results are not satisfactory. Machine learning methods have emerged to effectively model just such complex data. The classical machine learning methods used in this field include decision trees [5], k-nearest neighbors algorithm [6], and support vector regression (SVR) [7]. Nevertheless, they are unable to capture large-scale traffic as well as multi-featured spatio-temporal data and rely heavily on the processing of features set by technologists for the problem under study in the domain.

To better cope with such problems, a large number of researchers have turned to deep learning methods for analysis and to capture different feature variables by designing new architectures [8, 9]. Many deep learning models have achieved excellent results in different research areas, such as natural language processing [10], computer vision [11], recommender systems [12], and speech recognition [13]. In research in the field of transportation, the problem of predicting nonlinear traffic flows can be solved using classical neural network models. For example, long short-term memory (LSTM) models and gated recurrent units (GRU) models, which specialize in sequence prediction tasks, convolutional neural network (CNN) models, which are commonly used for image prediction, and stacked autoencoder (SAE), which extracts high-dimensional features from stacked structures. While these models can capture nonlinear correlations, they do not adequately account for spatio-temporal dependence. In recent years, more researchers have focused on how to better capture spatio-temporal dependence and heterogeneity, for instance, the modeling of traffic flow sequences by migrating machine translation in natural language processing [14] and capturing spatial correlations by using graph convolution derived from graph theory and signal processing [15]. The encoder-decoder framework is an end-to-end deep learning architecture that is also often used as a framework to handle sequence prediction tasks. This architecture also shows excellent performance when tackling tasks related to natural language processing [16].

In this paper, we propose an attention encoder-decoder dual graph convolutional network with time-series correlation (AED-DGCN-TSC) for multi-step traffic flow prediction, which can capture long series correlation with spatio-temporal characteristics through modules for multi-step prediction of future traffic characteristics. The model can process graph signals without relying on the fixed topology of the original traffic network, while introducing the idea of decomposition to aggregate similar subsequences.

The general structure in this paper is as follows: Section 2 shows the related research work. Section 3 introduces the concepts and definitions related to traffic forecasting tasks. Section 4 outlines the general framework and component details of our model AED-DGCN-TSC. The experiments and analysis of the proposed model and the baseline model are in Section 5, while the conclusions are presented in Section 6.

2. Related Work

In this section, we present the existing literature related to traffic flow prediction and compare the advantages and shortcomings of previous studies.

2.1. Graph Convolutional Network. In real life, there is a lot of data stored in the form of graphs. In traffic forecasting tasks, data are often processed as raster data or graph data, which facilitates easier organization for researchers. Examples include drop-off and pick-up points for cab trajectories, OD matrices, and bicycle stops. The classical convolution model can effectively extract local information from the data, but it requires the data to be strictly standard grid data. The graph convolutional neural network applies the idea of convolution to the graph structure. The graph data contain the topology represented by the adjacency matrix and the graph signal represented by the feature matrix. The graph convolution is a highly automated end-to-end learning, where both attribute information and structure information are learned simultaneously during the training process. Researchers have used such features to propose a series of models for the task of graphically structured data [17–19]. Fu et al. performed classification tasks on graph data using graph convolution [20]. Han et al. modelled the data from the proximity, day, and week perspectives, respectively, using a multilayer graph convolutional neural network overlay structure [21]. To fully capture the spatio-temporal characteristics of short-term traffic flow, Han et al. proposed an AST-GCN-LSTM model, in which LSTM is used to extract the features of temporal structure and combined with GCN to obtain the spatial characteristics [22]. Zhu et al. performed data fusion using a belief rule base (BRB) to obtain new traffic flow data and then uses a recurrent neural network (RNN) and GCN models to obtain the temporal correlation of traffic flows [23]. Wang et al. proposed the AST-MAGCN model to design multi-graph adversarial neural networks (GAN) to automatically obtain spatio-temporal states and spatio-temporal dependencies [24].

2.2. Traffic Forecasting Tasks. As electronics manufacturing technology becomes more and more sophisticated, it has given computers the ability to perform a large number of calculations in a short period. Under this environment, traffic prediction research also has more possibilities. With the advance of artificial intelligence technology, deep learning has turned out to be the tool of choice for many researchers. Chen et al. introduced multiple signal decomposition methods to denoise the traffic flow data, after which LSTM is introduced to complete the prediction task [25]. Abduljabbar et al. proposed a short-term traffic prediction model using unidirectional and bidirectional LSTM neural networks and demonstrates that its model can be used to predict speed and traffic over multiple prediction horizons [26]. Li et al. proposed a specific module to eliminate the differences between cycle data while modeling the dynamic temporal and spatial correlations caused by different traffic patterns between roads [27].

Improvement of traffic flow sequence prediction is achieved by introducing the attention mechanism [28]. Zhang et al. proposed a short-term traffic flow prediction model based on a temporal convolutional neural network (TCN) optimized by a genetic algorithm (GA) to improve the prediction accuracy of the TCN neural network by an optimization algorithm [29]. Ma et al. proposed a new capsule network (CapsNet) to extract the spatial features of the traffic network and use the nested LSTM structure to capture the time dependence of different granularity in the

traffic sequence [30]. Guo et al. added the attention mechanism module to the spatio-temporal graph convolution to capture dynamic spatio-temporal features in traffic data [31]. Zhao et al. developed a T-GCN model that combines graph convolution with recurrent neural networks to solve traffic prediction problems for urban road networks [32]. To model the hidden spatial dependencies, Wu et al. proposed an adaptive adjacency matrix structure to model the spatial properties between different network nodes [33]. Song et al. proposed a spatial-temporal synchronous graph convolutional network that aggregates temporal and spatial relationships on adjacent time steps to capture complex local features [34]. Li and Zhu then developed the STFGNN model to capture the global information that the STSGCN model ignores when capturing spatio-temporal features [35]. Recently, to extend the capability of sequence modeling, the encoder-decoder architecture has shown powerful capabilities. The encoder-decoder-based model is a generic end-to-end framework for sequence data processing that typically uses an encoder to encode the input sequence into a fixed dimensional vector and then decodes the target sequence from that vector as a prediction.

3. Preliminaries

We use $G = (V, E, \mathbf{A})$ to represent the traffic road network that contains the topology information of the road network, where $V = v_1, v_2, \ldots, v_N$ is a set of vertices represented by the sensor collecting the data, N denotes the number of vertices, E denotes the set of edges, $\mathbf{A} \in R^{N \times N}$ denotes the adjacency matrix. As shown in Figure 1, the blue circles in the figure indicate the nodes; the green lines represent the correlation of adjacent time steps of the current node; the brown lines indicate the correlation of adjacent time steps of neighboring nodes; the black lines indicate the correlation of neighboring nodes in the same time step.

Graph signal matrix $X_G \in \mathbb{R}^{N \times F}$ denotes the all graph signal on graph G, which contains the feature matrix of the corresponding nodes in the graph. F is the number of attribute features. The feature matrix can not only contain information such as flow and speed but also introduce external factors such as weather and POIs. Each node has its own feature matrix, for example, $F_2^{(v_3)}$ denotes the 2-th feature of node 3.

4. Methodology

Adequate consideration of the spatio-temporal characteristics of traffic flow and the evolutionary process of traffic node characteristics to the better establishment of spatio-temporal dependencies on long sequences. The framework structure of the AED-DGCN-TSC is illustrated in Figure 2. The framework is composed of an encoder, a decoder, and an attention mechanism in between. The encoder component employs the specific components as the basis for generating fixed-length, high-dimensional spatio-temporal feature vectors by graph structure data as input. The attention mechanism is used to augment the decoder by enabling it to focus on more details of the input information.

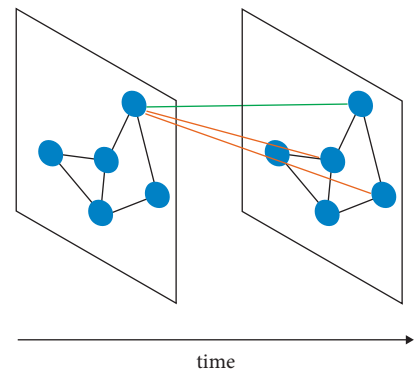

Figure 1: The structure of road network graph.

The decoder generates the traffic prediction scenarios of multiple future time steps from a representation vector of spatio-temporal features.

4.1. Time-Series Correlation Module. In the study of time series, the use of similar sequences to enhance the predictive power is quite effective [36]. As shown in Figure 3, we design a time-series correlation module (TSCM) with a tandem-connected auto correlation mechanism to enhance the utilization of time subsequences. The module finds period-based correlations by calculating sequence correlations and aggregates similar subsequences by delayed aggregation.

By studying the similarity of time series, hidden patterns can be better explored. The method of calculating the similarity of time series is often used in sequence clustering and feature extraction. The similarity of sample series is multi-faceted; it can be expressed as the relationship between the time series close to each other in the spatial distance, and it can also be expressed as having similar shapes. There are many methods to measure the similarity between different time series, and the common ones include Euclidean distance, Manhattan distance, Minkowski distance, pinch cosine, information entropy, and dynamic time warping (DTW).

Within repeated periods, the same phase positions between periods usually behave similarly. We denote the traffic flow time series by $\{X_t\}$. The time-series correlation $\mathscr{R}(\tau)$ is defined as follows:

$$\mathscr{R}(\tau) = \lim_{L \longrightarrow \infty} \frac{1}{L} \sum_{t=0}^{L-1} X_t X_{t-\tau}, \tag{1}$$

where $\mathscr{R}(\tau)$ portrays the time-delayed similarity of the time series X_t to the time series $X_{t-\tau}$ and takes it as the unnormalized confidence for estimating the period length τ. Then, we get multiple possible period lengths as our alternative length schemes. The corresponding time series similarity of the top k length schemes with the highest confidence level is selected for weighting.

Based on the consideration that periods are delayed, the time delay aggregation block is designed to roll the k time delay sequences $\tau_1, \tau_2, \ldots, \tau_k$ of the current time slice. As shown in Figure 4, the time delay aggregation block can aggregate similar time subsequences and implement

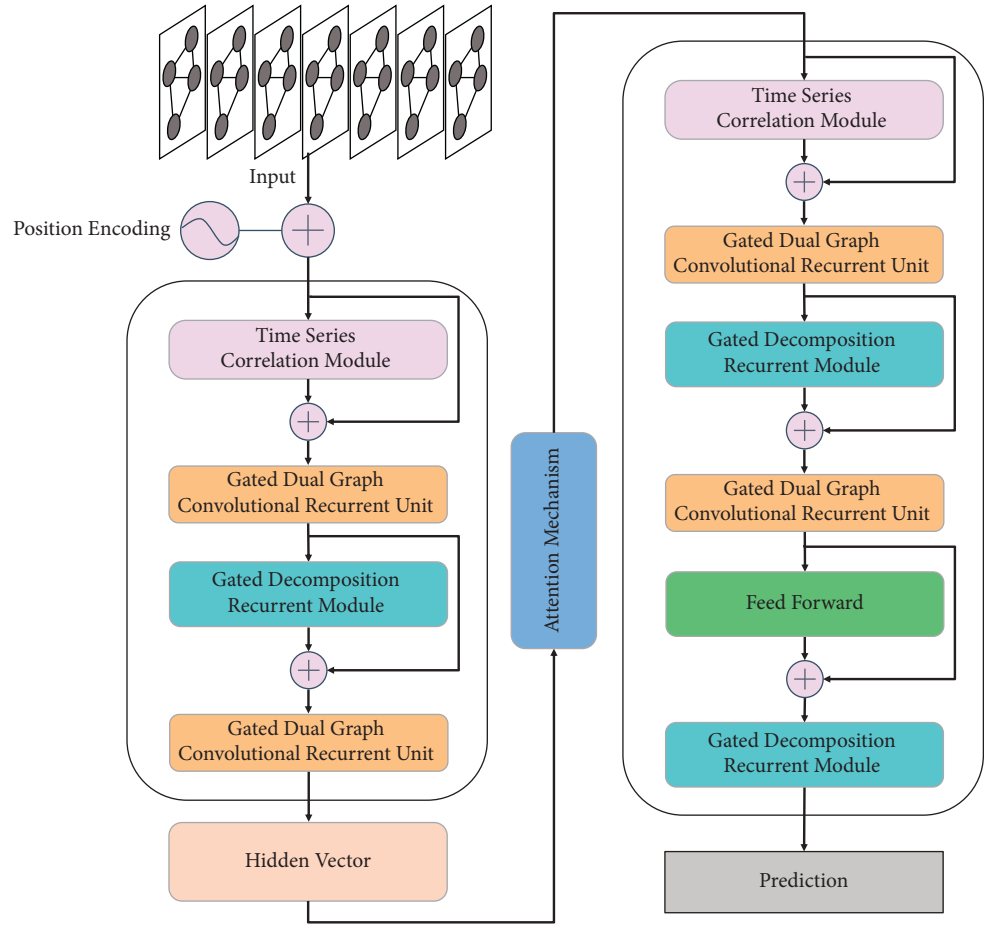

Figure 2: The framework of AED-DGCN-TSC.

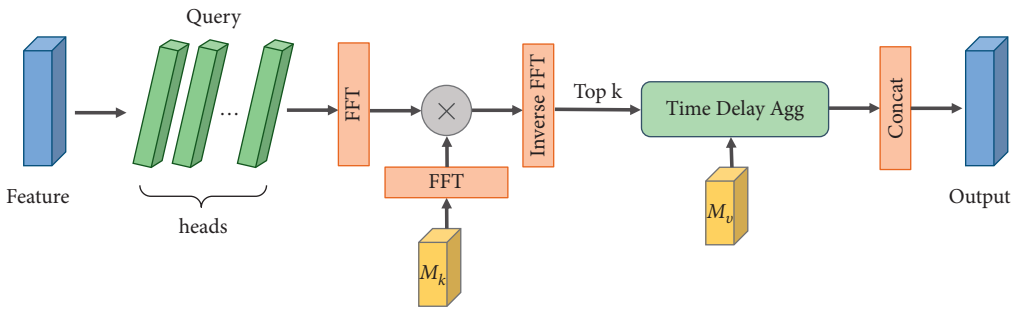

Figure 3: The structure of TSCM.

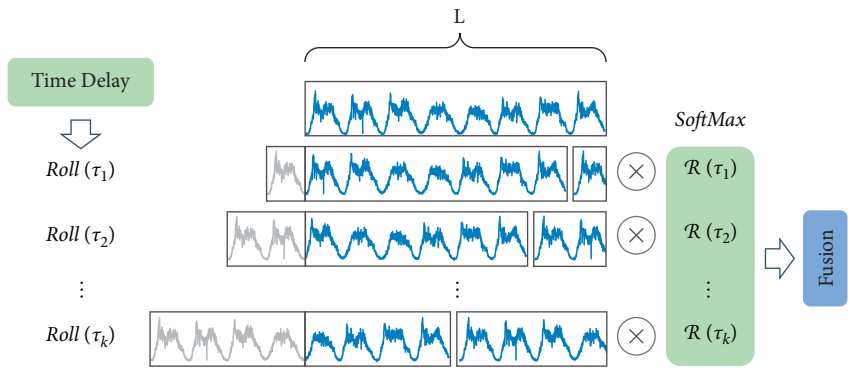

Figure 4: Time delay aggregation.

alignment operations by softmax function. Inspired by the external attention mechanism [37], for the single head case of length L and time series X, we project the input to query Q and two different memory units M_k and M_v as the key and value. The time-series correlation mechanism is defined as formulas (2)–(4):

$$\tau_1, \cdots, \tau_k = \underset{\tau \in \{1,\ldots,L\}}{\arg \operatorname{Top} k}\left(\mathscr{R}_{Q,M_k}(\tau)\right), \tag{2}$$

$$\begin{aligned}&\widehat{\mathscr{R}}_{Q,M_k}(\tau_1), \ldots, \widehat{\mathscr{R}}_{Q,M_k}(\tau_k)\\ &\quad = \operatorname{Soft\,Max}\left(\mathscr{R}_{Q,D_k}(\tau_1), \ldots, \mathscr{R}_{Q,M_k}(\tau_k)\right),\end{aligned} \tag{3}$$

$$\begin{aligned}&\text{Time Series Correlation}\left(Q, M_k, \mathrm{V}\right)\\ &\quad = \sum_{i=1}^{k} \operatorname{Roll}\left(\mathscr{M}_{\blacklozenge}, \tau_k\right)\widehat{\mathscr{R}}_{Q,M_k}(\tau_k),\end{aligned} \tag{4}$$

where arg Top k denotes the first k parameters obtained according to $\mathscr{R}_{Q,M_k}(\tau_k)$. $\mathscr{R}_{Q,M_k}(\tau)$ is the sequence similarity between Q and M_k through a period length of τ. In determining the size of k, we calculate it using $k = c \times \log L$, where c is a hyper-parameter. $\operatorname{Roll}(X, \tau)$ represents the transformation operation of the sequence X with time delay τ, during which the elements that were moved beyond the first position will be reintroduced at the last position. The multi-head version enriches the capabilities of the model and stabilizes the training process with the following formula:

$$\begin{aligned}h_i &= \text{Time Series Correlation}\left(F_i, M_k, M_v\right)F_{\text{out}}\\ &= \operatorname{MultiHead}\left(F, M_k, M_v\right)\\ &= \operatorname{Concat}\left(h_1, \ldots, h_H\right)W_o,\end{aligned} \tag{5}$$

where F denotes the feature matrix of the input, h_i is the i-th head, and H denotes the number of heads. W_o is used to obtain a linear transformation matrix with consistent dimensional of the input and output. M_k and M_v are shared memory units with different heads.

The Fourier transform accomplishes the conversion of time series in the time and frequency domains. The traditional Fourier transform is applied to continuous functions, while a discrete version of the Fourier transform will be used since our data is discrete. As shown in formulas (6) and (7), for a given time series X, we will apply the fast Fourier transform (FFT) to calculate $\mathscr{R}(\tau)$, which is an algorithm for computing the discrete Fourier transform (DFT) and its inverse (IDFT).

$$\mathrm{S}(f) = \mathscr{F}\left(\mathrm{X}_t\right)\mathscr{F}^*\left(\mathrm{X}_t\right) = \int_{-\infty}^{\infty} \mathrm{X}_t e^{-i2\pi t f}\,\mathrm{d}t \overline{\int_{-\infty}^{\infty} \mathrm{X}_t e^{-i2\pi t f}\,\mathrm{d}t}, \tag{6}$$

$$\mathscr{R}(\tau) = \mathscr{F}^{-1}\left(\mathrm{S}(f)\right) = \int_{-\infty}^{\infty} \mathrm{S}_{\mathrm{xx}}(f) e^{i2\pi f \tau}\,\mathrm{d}f, \tag{7}$$

where $\mathscr{F}$ denotes FFT and $\mathscr{F}^{-1}$ denotes its inverse. $\mathscr{F}^*$ denotes the conjugate operation of the FFT, and $\mathrm{S}(f)$ is the frequency domain. All the lag values can be calculated by one FFT operation.

4.2. Dual Graph Convolution Operation. Effectively capturing spatio-temporal dependencies is an important issue in modeling spatio-temporal data of traffic flow. Although the convolutional approach allows the extraction of spatial features, it is not applicable to data beyond images. The development of graph convolutional networks, which are extensions of convolutional neural networks to graph structures, has benefited from the extension of the underlying theory of graph signal processing to graph convolutional neural networks. Graph convolution can perform many tasks related to graphs, such as node classification, graph classification, and link prediction. When dealing with a specific task, it is often divided into two types of views, a null domain view or a frequency domain view, depending on the understanding of the graph filter.

GCN can determine the location relationship between the central node and the surrounding nodes, while encoding the network structure and attributes of the road, so that the topology and interrelationships in the road network structure can be obtained effectively. As a variant of graph neural network, the graph attention network shown in Figure 5 performs the aggregation operation on neighbor nodes through the attention mechanism, and the weights of different neighbor nodes are assigned adaptively. Unlike the static weights of standard graph convolution, using dynamic graph attention captures interactions between nodes more effectively:

$$\mathbf{f}_i^{'} = \alpha_{i,i}\mathbf{W}\mathbf{f}_i + \sum_{j \in \mathrm{N}(i)} \alpha_{i,j}\mathbf{W}\mathbf{f}_j, \tag{8}$$

where $\mathbf{f}_i$ denotes the feature of vertice i and $\mathbf{W}$ denotes learnable parameter. $\alpha_{i,i}$ is the attention coefficients, which can be calculated as follows:

$$\alpha_{i,j} = \frac{\exp\left(\mathbf{a}^{\top}\operatorname{LeakyReLU}\left(\mathbf{W}\left[\mathbf{f}_i \parallel \mathbf{f}_j\right]\right)\right)}{\sum_{k \in \mathrm{N}(i) \bigcup \{i\}} \exp\left(\mathbf{a}^{\top}\operatorname{LeakyReLU}\left(\mathbf{W}\left[\mathbf{f}_i \parallel \mathbf{f}_k\right]\right)\right)}, \tag{9}$$

where $\mathbf{a}$ and $\mathbf{W}$ are learnable parameters. $[\mathbf{f}_i \parallel \mathbf{f}_j]$ is the concatenated features at vertices i and j.

As shown in Figure 6, the basic principle of the dual graph is to map the edges in the primal graph to the nodes in the dual graph. Relatively, we refer to the primal graph structure $G = (V, E, \mathbf{A})$ before the transformation as the primal graph. The dual graph of G is denoted as $\widetilde{G} = (\widetilde{V}, (\widetilde{E}, E), \widetilde{A})$, $\widetilde{V}$ and $\widetilde{E}$ denote the set of vertices and the set of edges after conversion to a dual graph, respectively, and $\widetilde{A}$ is the adjacency matrix that can be obtained from the conversion of $\mathbf{A}$. The specific construction principle is as follows: each dual vertex $(i, j) \in \widetilde{V}$ is transformed from the corresponding edge of the primal graph $(i, j) \in E$, and if two dual vertices (i, j), $(i', j') \in \widetilde{V}$ share a common direction and at least one endpoint in G, they will form an edge between them.

Based on the idea of dual graph, the feature matrix of the dual graph $\widetilde{G}$ corresponding to the feature matrix $X_G \in \mathbb{R}^{N \times F}$ of the primal graph G is $X_G \in \mathbb{R}^{\mathrm{N} \times 2F}$. The dual

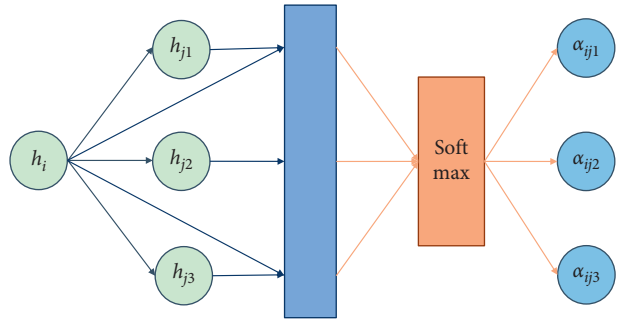

Figure 5: Graph attention mechanism.

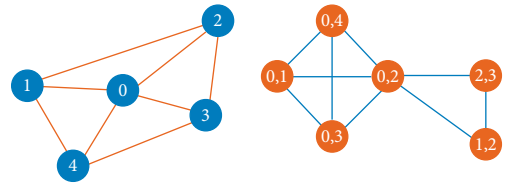

Figure 6: Primal graph (a) and dual graph (b).

vertex features $\tilde{\mathbf{f}}_i = [\mathbf{f}_i, \mathbf{f}_j]$ constructed by concatenating the respective primal vertex features $(i, j) \in E$. For each edge $\{i, j\}$ of the undirected edge in the dual graph, we construct two nodes of $(i, j)(j, i) \in \tilde{V}$. During construction, we connect a single dual node (i, j) with all the nodes corresponding to the edges pointing to i or deviating from j. Consequently, we can apply the dynamic graph attention mechanism to the corresponding feature matrix on the dual graph:

$$\tilde{f}'_{ij} = \mathrm{ReLU}\left(\sum_{r\in N(i)} \tilde{\alpha}_{ij,ir}\tilde{W}\tilde{f}_{ir} + \sum_{t\in N(j)} \tilde{\alpha}_{ij,tj}\tilde{W}\tilde{f}_{tj}\right),$$
$$\tilde{\alpha}_{ij,ik} = \frac{\exp\left(\mathbf{a}^{\top}\gamma\left(\tilde{W}\left[\mathbf{f}_{ij} \parallel \mathbf{f}_{ik}\right]\right)\right)}{\sum_{r\in N(i)}\exp\left(\mathbf{a}^{\top}\gamma\left(\tilde{W}\left[\mathbf{f}_{ij} \parallel \mathbf{f}_{ir}\right]\right)\right) + \sum_{r\in N(i)}\exp\left(\mathbf{a}^{\top}\gamma\left(\tilde{W}\left[\mathbf{f}_{ij} \parallel \mathbf{f}_{tj}\right]\right)\right)}, \tag{10}$$

where $\tilde{f}'_{ij} \in \mathbb{R}^{\tilde{F}}$ is the feature output matrix of the dual vertices. $\tilde{W}$ is the learnable parameter. $\tilde{\alpha}_{ij,ir}$ is the dynamic dual attention scores and $\mathbf{a}^{\top}$ is a vector of $2\tilde{F}$ dimensions. γ is the Leaky ReLU activation function. For the primal graph, we complete the aggregation operation using the following formulas:

$$\mathbf{f}'_i = \mathrm{ReLU}\left(\alpha_{i,i}\mathbf{W}\mathbf{f}_i + \sum_{r\in N(i)} \alpha_{i,j}\mathbf{W}\mathbf{f}_j\right),$$
$$\alpha_{i,j} = \frac{\exp\left(\mathbf{a}^{\top}\mathrm{LeakyReLU}\left(\mathbf{W}\left(\tilde{f}'_{ij}\right)\right)\right)}{\sum_{k\in N(i)\cup\{i\}}\exp\left(\mathbf{a}^{\top}\mathrm{LeakyReLU}\left(\mathbf{W}\left(\tilde{f}'_{ik}\right)\right)\right)}, \tag{11}$$

where $\mathbf{f}'_i \in \mathbb{R}^{F'}$ denotes the output features at primal vertex i. $\mathbf{W}$ is the learnable parameter and $\alpha_{i,j}$ is the primal attention score. The above operation process is called dual graph convolution operation (DGCO), and the spatial properties are discovered by applying graph attention mechanism in the primal graph and the dual graph [38].

4.3. Gated Dual Graph Convolutional Recurrent Unit. Considering the requirement of integrated capture of spatio-temporal correlations, this study proposes a gated dual graph convolutional recurrent unit (GDGCRU), which consists of a DGCO-GRU that embeds the DGCO into the GRU and an adaptive graph convolutional recurrent network (AGCRN) [39] that is used to represent the implicit spatial relationships. Its structure is shown in Figure 7.

RNN is a model of recurrent feedback with a complex composition structure. LSTM is a modified RNN model with an internal structure that controls the storage of information through a gating mechanism, including input gates, forgetting gates, and output gates. GRU retains only the structure of update gate and reset gate, which is also suitable for predicting long-time sequences. Benefiting from its simple structure and fast computation, GRU is commonly used in tasks of speech recognition and machine translation. To enhance the acquisition of spatial properties in sequence modeling, we replace the linearly connected layers in GRU

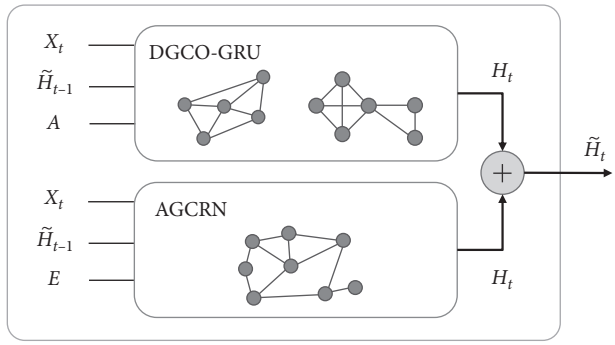

Figure 7: The structure of the GDGCRU.

with our proposed graph convolution structure following the ConvLSTM model structure [40]. In DGCO-GRU, we use a similar gating mechanism structure as GRU. The reset gate, update gate, and hidden state are defined as follows:

$$\begin{aligned} u_t &= \sigma\left(W_u{}^*\left[DGCO\left(A, X_t\right), h_{t-1}\right] + b_u\right), \\ r_t &= \sigma\left(W_r{}^*\left[DGCO\left(A, X_t\right), h_{t-1}\right] + b_r\right), \\ c_t &= \tan h\left(W_c{}^*\left[DGCO\left(A, X_t\right), \left(r_t{}^* h_{t-1}\right)\right] + b_c\right), \\ h_t^d &= \left(u_t{}^* h_{t-1} + \left(1 - u_t\right)^* c_t\right). \end{aligned} \tag{12}$$

Among them, h_{t-1} denotes the hidden state at moment $t-1$ and X_t denotes the traffic flow characteristics at the current moment. W and b denote the learnable parameters corresponding to each component. u_t indicates the update gate, which is used to keep the information valid for the previous state. r_t represents the reset gate, which controls the extent to which the status information from the previous moment is ignored. If necessary, information that is not relevant to the prediction can be ignored. c_t indicates the memory content of the current moment and h_t^d denotes the hidden state at the current moment. DGCO-GRU constitutes the current state by combining the current input information with the implied state of the previous moment. The recognition of spatio-temporal properties in the current state is enhanced by the involvement of graph convolution.

Nevertheless, most of the current research assumes that the structure of relationships between nodes is determined and then learns spatial dependencies based on a fixed graph structure. To enhance the representation of features by implicit graph structures, we introduced the AGCRN model [39] to capture implicit spatial properties that are not easily captured by DGCO-GRU. Such a prefixed structure dilutes the importance of the hidden graph structure in the learning process. The AGCRN can dynamically discover the relationships between nodes from the data:

$$\begin{aligned} \widehat{\mathbf{A}} &= \mathrm{Softmax}\left(\mathrm{ReLU}\left(\mathbf{E}\mathbf{E}^{\mathrm{T}}\right)\right), \\ z_t &= \sigma\left(\widehat{\mathbf{A}}\left[X_t, h_{t-1}\right]\mathbf{E}\mathbf{W}_{\mathbf{z}} + \mathbf{E}\mathbf{b}_{\mathbf{z}}\right), \\ r_t &= \sigma\left(\widehat{\mathbf{A}}\left[X_t, h_{t-1}\right]\mathbf{E}\mathbf{W}_r + \mathbf{E}\mathbf{b}_r\right), \\ \widehat{h} &= \tan \mathrm{h}\left(\widehat{\mathbf{A}}\left[X_t, r_t \odot h_{t-1}\right]\mathbf{E}\mathbf{W}_{\widehat{h}} + \mathbf{E}\mathbf{b}_{\widehat{h}}\right), \\ h_t^a &= \mathbf{z} \odot h_{t-1} + (1 - \mathbf{z}) \odot \widehat{h}_t, \end{aligned} \tag{13}$$

where $\mathbf{E}$ denotes the adaptive adjacency matrix implemented by the learnable node embedding, which is randomly initialized and progressively studies internode correlations. X_t and h_t^a denote the input and output of time step t, respectively. ReLU and Softmax both are activation functions, and they are utilized to weaken connections. $\mathbf{W_z}$, $\mathbf{W_r}$, $\mathbf{W}_{\widehat{h}}$, $\mathbf{b_z}$, $\mathbf{b_r}$, and $\mathbf{b}_{\widehat{h}}$ are learnable parameters in AGCRN. Finally, the output states of DGCO-GRU and AGCRN are summed as the output of GDGCRU:

$$h_t = h_t^d + h_t^a. \tag{14}$$

4.4. Gated Decomposition Recurrent Module. When collecting the information related to traffic characteristics, the collector is often inevitably disturbed by external factors, such as weather conditions and the state of the sensor itself. The raw information collected in this way contains noisy data that may interfere with the model. Decomposition of the data using the idea can effectively filter this noisy information, thus more accurately identifying features from the data. In the research on traffic flow analysis, besides considering traffic flow as a whole for traffic flow prediction, another approach is to consider traffic flow as a combination of multiple rule components generated by travelers, such as private cars and buses. These rule components are also affected by external factors, such as traffic accidents and bad weather.

The complex characteristics of these time dimensions are often difficult to capture. To facilitate series analysis, we divide the time series into a trend-cycle component and a seasonal component, which reflect the long-term development and seasonality of the series, as illustrated in Figure 8. The decomposition process is as follows:

$$\begin{aligned} \mathrm{X}_{\mathrm{trend}} &= \mathrm{AvgPool}\left(\mathrm{Padding}\left(\mathrm{X}\right)\right), \\ \mathrm{X}_{\mathrm{season}} &= \mathrm{X} - \mathrm{X}_{\mathrm{t}}, \end{aligned} \tag{15}$$

where $\mathrm{X}_{\mathrm{trend}}, \mathrm{X}_{\mathrm{season}}$ denote the seasonal and extracted trend-cyclical components, respectively. AvgPool and Padding denote average pooling and padding operations, respectively.

As shown in Figure 9, we propose the gated decomposition recurrent module (GDRM), which captures the spatio-temporal properties using the characteristics of

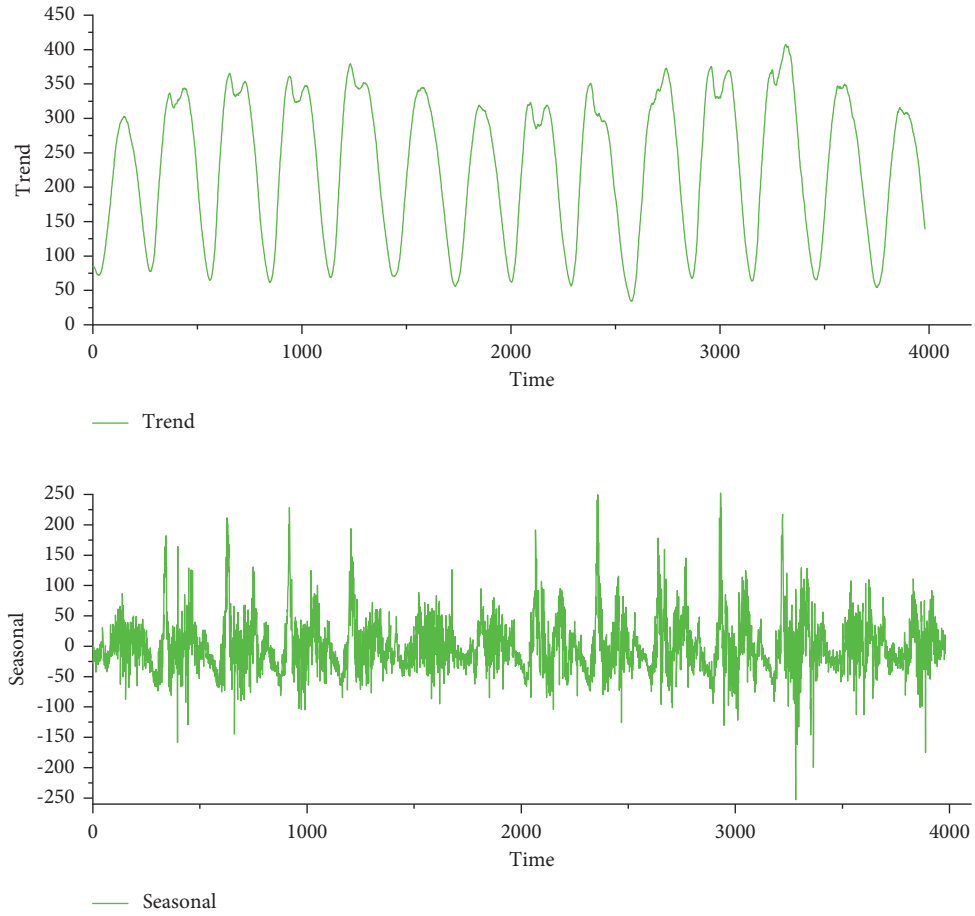

Figure 8: The trend component and seasonal component.

decomposition. After decomposing the input into trend and seasonal terms, the temporal dependence is captured by ConvGRU, respectively. The ConvGRU calculation formula is shown in formula (15). Eventually, the spatial information is aggregated from the spatial dimension by the first-order Chebyshev graph convolution as shown in formula (17):

$$\begin{aligned} Z_t &= \sigma\left(W_{xz} * X_t + W_{hz} * H_{t-1}\right), \\ R_t &= \sigma\left(W_{xr} * X_t + W_{hr} * H_{t-1}\right), \\ H_t^{'} &= f\left(W_{xh} * X_t + R_t \circ \left(W_{hh} * H_{t-1}\right)\right), \\ H_t &= \left(1 - Z_t\right) \circ H_t^{'} + Z_t \circ H_{t-1}, \end{aligned} \tag{16}$$

where Z_t denotes the reset gate and R_t denotes the update gate. W denotes the learnable parameter and $*$ stands for the convolution operator.

$$\mathbf{Z} = \left(\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\right)\mathbf{X}\boldsymbol{\Theta} + \mathbf{b}, \tag{17}$$

where $\mathbf{I}$ denotes a unit matrix of the same size as the adjacency matrix. Both Θ and $\mathbf{b}$ are parameters that can be learned in the train. $\mathbf{D}$ is the diagonal degree matrix.

4.5. Encoder-Decoder with Attention. The encoder-decoder structure is commonly used in language modeling by encoding the encoded character sequence into a high-level unified semantic vector, and decoding is done by a decoder when used. The entire encoder-decoder model structure consists of two main components: the encoder and the decoder. Usually, encoders and decoders use LSTM or GRU as their composition, and they model the time-series data by interlinking the internal nodes during sequence modeling. Nevertheless, this structure focuses on capturing specific patterns in the correlation of temporal dimensions and ignores the interplay between time and space. To enhance the overall ability to capture spatio-temporal characteristics, the model proposed in this study does not use the GRU or LSTM as encoder and decoder but uses specific components to model spatio-temporal information. The decoder component makes predictions of traffic flow sequence features for future time steps based on the semantic vectors trained by the encoder.

Not only above, but the encoder-decoder model also needs to compress the contextual information of the original message in the encoder into a fixed-length vector. This makes the model limited in handling long sequence data. As the length of the input sequence increases, the performance of the traditional encoder-decoder may find a rapid degradation. To address this issue, we use an attention mechanism to enhance the contribution of the original sequence features to the model. The attention mechanism is calculated as follows:

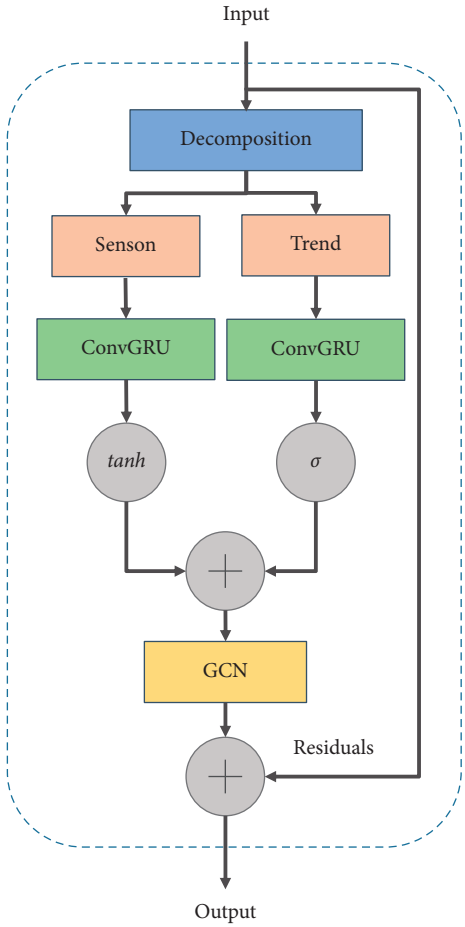

FIGURE 9: The structure of the gated decomposition recurrent module.

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j,$$

$$\alpha_{i,j} = \frac{\exp\left(e_{i,j}\right)}{\sum_{k=1}^{T} \exp\left(e_{i,k}\right)}, \tag{18}$$

$$e_{i,j} = e\left(s_{i-1}, h_j\right)$$

The symbol c_i denotes a fixed-length vector trained in encoder that is obtained by multiplying and summing the product of each hidden state h and the corresponding attention weight α. The attention weight $\alpha_{i,j}$ is the attention weight score of the input data at the time slice with time step j for the future predicted time slice with time step i. The vector $e_{i,j}$ used to align the sequence is normalized by SoftMax.

In summary, the proposed encoder-decoder traffic flow prediction model can match the predicted data well with the roads and can predict the traffic situation of the whole road network, which means that the encoder-decoder of the proposed traffic flow deep learning framework can effectively learn the deep features and use the decomposition technique to explore the hidden features.

4.6. Loss Function. In order to make the difference between the multi-step prediction and the actual value as small as possible, we use $L1$ loss as the loss function. L_{reg} is used as a normalization term to prevent overfitting and λ is the hyperparameter.

$$\text{loss} = \sum_{i=t+1}^{i=t+T'} \| Y_i - Y_i^{'} \| + \lambda L_{\text{reg}}. \tag{19}$$

5. Experiments and Analysis

In this section, the performance of the model is evaluated based on two real data using the Caltrans Performance Measurement System (PeMS) dataset [41].

5.1. Data Description. PEMSD4 and PEMSD8 were collected in two areas of California. The PeMS system measures California's highway traffic characteristics, such as speed, volume, and occupancy, every 30 seconds in real time. We aggregate the flow data into 5 minutes and use it as a time step. We select speed features to validate the model. The PEMSD4 dataset is the cab trajectory in Shenzhen between January 1 and February 28, 2018. This dataset was selected from 307 collection points as the study area, while the PEMSD8 dataset is composed of data collected from 170 devices from July 1 to August 31, 2016.

5.2. Baselines. We compare the following model with the proposed AED-DGCN-TSC model:

(i) HA: the historical average model is a method that uses historical data to take an average value as a prediction.

(ii) VAR [42]: vector autoregression is a time-series forecasting model that is valid for systems of interconnected time-series variables.

(iii) ARIMA: autoregressive integrated moving average model is one of the most common statistical models used to make time-series forecasts.

(iv) GRU [43]: gate recurrent unit is a model that preserves the valid features of the data through a gating mechanism.

(v) DCRNN [44]: diffusion convolutional recurrent neural network captures spatio-temporal properties using diffusion graph convolution in a sequence-to-sequence learning structure.

(vi) STGCN [45]: spatial-temporal graph convolutional network combines the ideas of graph convolution and convolutional networks to model spatio-temporal sequences.

(vii) ASTGCN: attention-based spatial-temporal graph convolutional networks capture spatial and temporal correlations using a spatial attention mechanism and a temporal attention mechanism.

5.3. Evaluation Metrics. We used three different evaluation metrics to measure the predictive effect of the model, such as root mean-squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). The three evaluation indicators are calculated as follows:

$$\begin{aligned} \mathrm{MAE} &= \frac{1}{N}\sum_{i\in T}|y_i - \widehat{y}_i|, \\ \mathrm{MAPE} &= \frac{100\%}{N}\sum_{i\in T}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|, \\ \mathrm{RMSE} &= \sqrt{\frac{1}{N}\sum_{i\in T}(y_i - \widehat{y}_i)^2}, \end{aligned} \tag{20}$$

where y is the actual data, $\widehat{y}$ represents the predicted value, and T is the length of the time dimension of the observed sample. The smaller the value derived from the above formula, the better the result is proven.

5.4. Experiment Settings. The AED-DGCN-TSC proposed in this study is implemented through Pytorch and Scikit-learn libraries. We use all datasets with 60% as training set, 20% as validation set, and the remaining 20% as test set. Models were trained in a Pytorch 1.8.1 and cuda 11.1 environment with an TeslaT4 rtx300 GPU 16G card. We train our model using Adam optimizer with learning rate 0.001 and the batch size is 16 and the training epoch is 200. The ConvGRU's input dimension in the GDRM structure is 12, the hidden layer size is set to (32, 16, 12), the convolution kernel is set to (3, 5, 3), and the number of cell layers is set to 3. The residual unit keeps features along the time and space dimensions, and a two-dimensional convolution with a convolution kernel of (1, 1)and a step size of (1, 1) is used. To make the model avoid overfitting, the dropout parameter is set to 0.05 during model training. Adding layer normalization processing to the long series model increases the stability of the model for small batches of data and dynamic networks. Larger filters tend to capture more information, but that does not mean the larger the filter size the better the model will be. In the comparison model experiments, the K value of Chebyshev polynomial in the ASTGCN model is set to 3, and the convolution kernel size is set to 64 for both the graph convolution layer and the temporal convolution layer. The optimal parameters for all baseline models are selected by adjusting the effect of the parameters verified on the validation set.

5.5. Experiment Results. To better compare the model performance, we compare the average results of the data predicted by the model for the next 12 time steps. The results of the experiment are listed in Tables 1 and 2. The proposed AED-DGCN-TSC model consistently outperformed the other baseline models on the two data sets. As can be seen from the table, the traditional statistical modeling methods such as HA, VAR, and ARIMA have certain limitations. Sequence modeling approaches such as GRU and DCRNN focus on capturing the properties of the temporal dimension and ignore the importance of the spatial properties. Compared with DCRNN, the proposed model has a 6.818% effect reduction in the RMSE metric and reduces the MAE metric by 13.846%. Although STGCN and ASTGCN use graph convolution to take spatio-temporal properties into account, the corresponding modeling methods are designed in both time and space separately, rather than combining spatio-temporal properties in an integrated manner. On both datasets, the proposed model reduces the RMSE metric by 10.690% and 8.896%, respectively, compared to the STGCN model. In the 12-step prediction experiment, the 12th step prediction increases 8.779% and 8.452% in the MAE and RMSE metrics, respectively, compared to the 1st step prediction. Since both the input and output of the graph convolution method are graph structures, the proposed AED-DGCN-TSC model with the advantage of such a feature is able to output the predicted values of all nodes in the road network. The predicted and true values of the speed corresponding to all nodes on both PeMSD4 and PeMSD8 datasets are shown in Figure 10. The AED-DGCN-TSC model proposed in this study incorporates the information of adjacent time steps and combines the encoder-decoder model with the gated recurrent structure to fully exploit the advantages of long sequence multi-step prediction and aggregation space properties. To demonstrate our model visually, the comparison histogram with ASTGCN and STGCN is shown in Figure 11.

The adaptive adjacency matrix represented by node embedding in graph convolution affects the effectiveness of capturing spatial information, and thus the size of the embedding dimension helps to better learn the information implicit in the graph structure. As shown in Figure 12, the performance when the embedding dimension from 2 to 20 is tested by MAE and RMSE metrics on the PEMSD4 dataset. As can be seen from the figure, the model performs best when the embedding dimension is set to 10. Both too small and too large node embedding dimensions affect the performance. Larger dimensional node embedding can contain more information, but choosing too large a dimension can be counterproductive and make the model difficult to fit. After comparing the effects of different embedding dimensions for the experiments on the two datasets, the best results were obtained for the PEMSD4 and PEMSD8 datasets by choosing 10 and 3, respectively.

5.6. Performance of Multi-Step Forecasting. Multi-step forecasting has a longer forecast range than single-step forecasting, which will give the traffic department enough time to dispatch the traffic [46, 47]. Multi-step forecasting can provide more reflective forecasts of future trends, such as a smooth increase or a steep decrease in the change of forecast values over a future period, which cannot be observed with a single forecast value obtained from single-step forecasting. Instead of predicting traffic flow characteristics for a single node, the proposed model has the ability to predict the future characteristics of multiple nodes in the dataset. The changes in the prediction performance of STGCN, ASTGCN, and AED-DGCN-TSC models as the prediction interval increases are

Table 1: Performance comparison of different baseline models predicting future data at 12 time steps on PEMSD4 dataset.

Metrics	HA	VAR	ARIMA	GRU	DCRNN	STGCN	ASTGCN	AED-DGCN-TSC
MAE	38.03	24.54	31.56	23.68	21.22	21.16	22.93	19.23
RMSE	59.24	38.61	54.21	39.27	33.44	34.89	35.22	31.16
MAPE	27.88%	17.24%	28.72%	16.44%	14.17%	13.83%	16.56%	12.32%

Table 2: Performance comparison of different baseline models predicting future data at 12 time steps on PEMSD8 dataset.

Metrics	HA	VAR	ARIMA	GRU	DCRNN	STGCN	ASTGCN	AED-DGCN-TSC
MAE	34.86	19.19	32.49	22.00	16.82	17.50	18.25	16.04
RMSE	52.04	29.81	48.39	36.23	26.36	27.09	28.06	24.68
MAPE	24.07%	13.10%	25.12%	13.33%	10.92%	11.29%	10.02%	10.28%

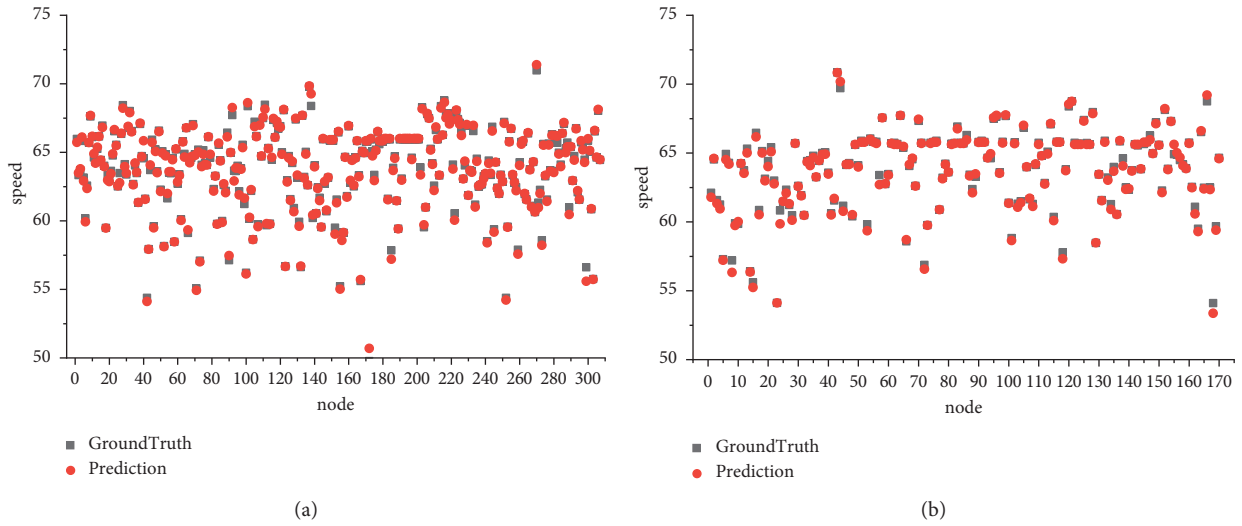

Figure 10: The real values and predicted values for all nodes on (a) PeMSD4 and (b) PeMSD8.

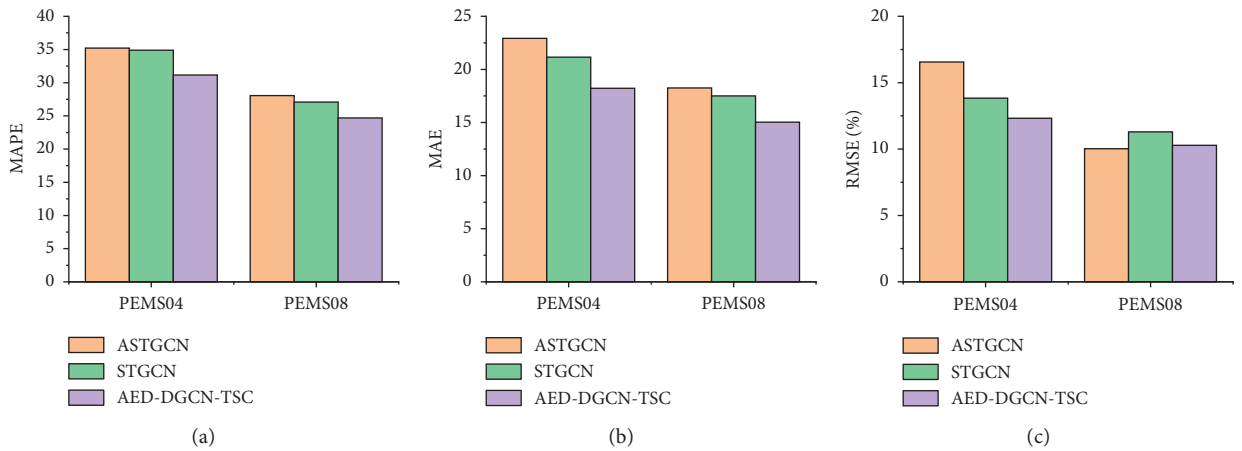

Figure 11: Comparison of the performance of ASTGCN, STGCN, and AED-DGCN-TSC with two datasets. (a) Comparison of MAE. (b) Comparison of RMSE. (c) Comparison of MAPE.

shown in Figure 13. The proposed model is compared with two models that mine spatio-temporal data information, and although the effect is slightly worse than ASTGCN in the first time interval, the prediction effect is more stable as the time interval increases, and finally, a good prediction result is achieved in the whole. Compared with ASTGCN, which applies temporal attention mechanism and spatial attention mechanism respectively, our model further explores the spatio-temporal pattern from the improvement of graph convolution and similar time series.

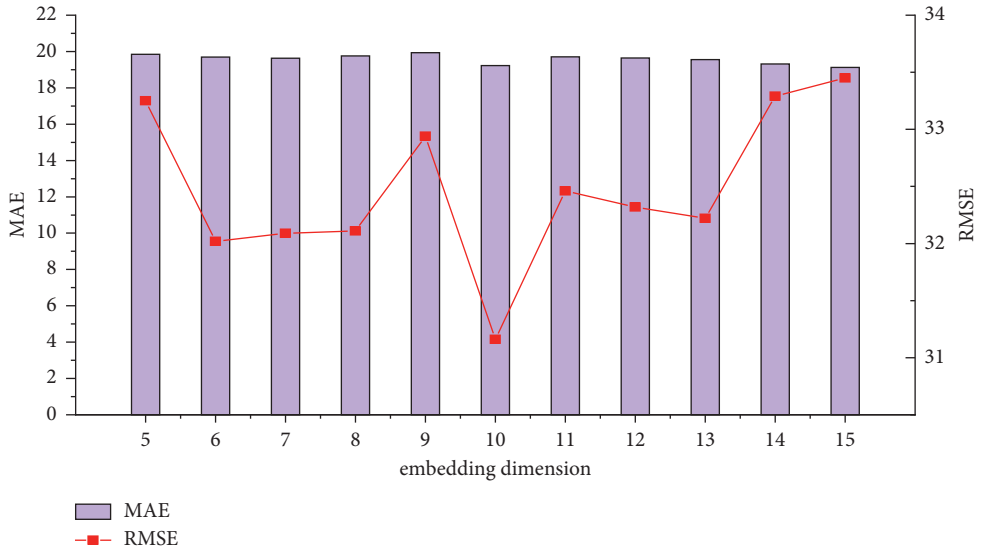

Figure 12: The effect of embedding dimension on the prediction performance.

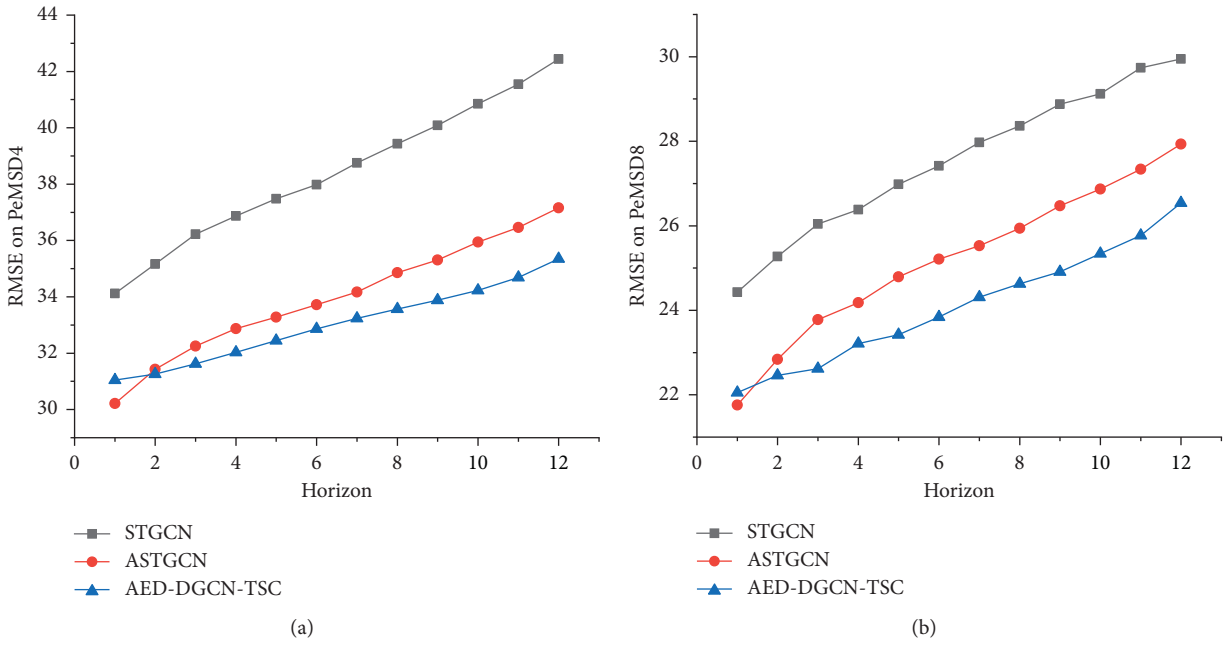

Figure 13: Prediction performance comparison on PeMSD4 and PeMSD8 datasets. (a) RMSE on PeMSD4. (b) RMSE on PeMSD8.

In order to compare the effectiveness of encoder-decoder structure, we subjected the AED-DGCN-TSC model to multi-step prediction experiments, and the results of step 6, step 12, and step 24 are listed in Table 3. From the data in the table, it can be seen that the encoder-decoder model combining the ideas of graph convolution and decomposition is more capable of capturing the spatio-temporal dependence of long sequences and can effectively perform multi-step prediction. As the prediction length increases, the prediction effect will gradually become worse, so it is not advisable to set too long prediction steps in solving practical problems.

5.7. Performance in Different Forecast Periods. Prediction experiments were conducted for each of the three characteristics of traffic flow, occupancy, and speed in the dataset. Figures 14 and 15 correspond to the experimental results of the two datasets, where the red lines represent the true values of the corresponding characteristics in the dataset, while the black lines represent the predicted values of the model output. In Figure 14, the RMSE indicators for the three traffic flow characteristics are 31.05, 0.01, and 1.92 for the PEMSD4 dataset, while the MAPE indicators reach 12.63%, 16.66%, and 1.94%, respectively. In Figure 15, the RMSE metrics for the three traffic flow characteristics on the

Table 3: Performance comparison of different baseline models predicting future data at 12 time steps on PEMSD8 dataset.

Metrics	RMSE			MAE			MAPE		
	6-step	12-step	24-step	6-step	12-step	24-step	6-step (%)	12-step (%)	24-step (%)
PEMSD4	31.88	33.49	37.41	19.85	20.94	22.95	12.93	13.63	15.03
PEMSD8	25.71	28.58	31.51	15.85	16.25	18.20	10.82	12.05	14.08

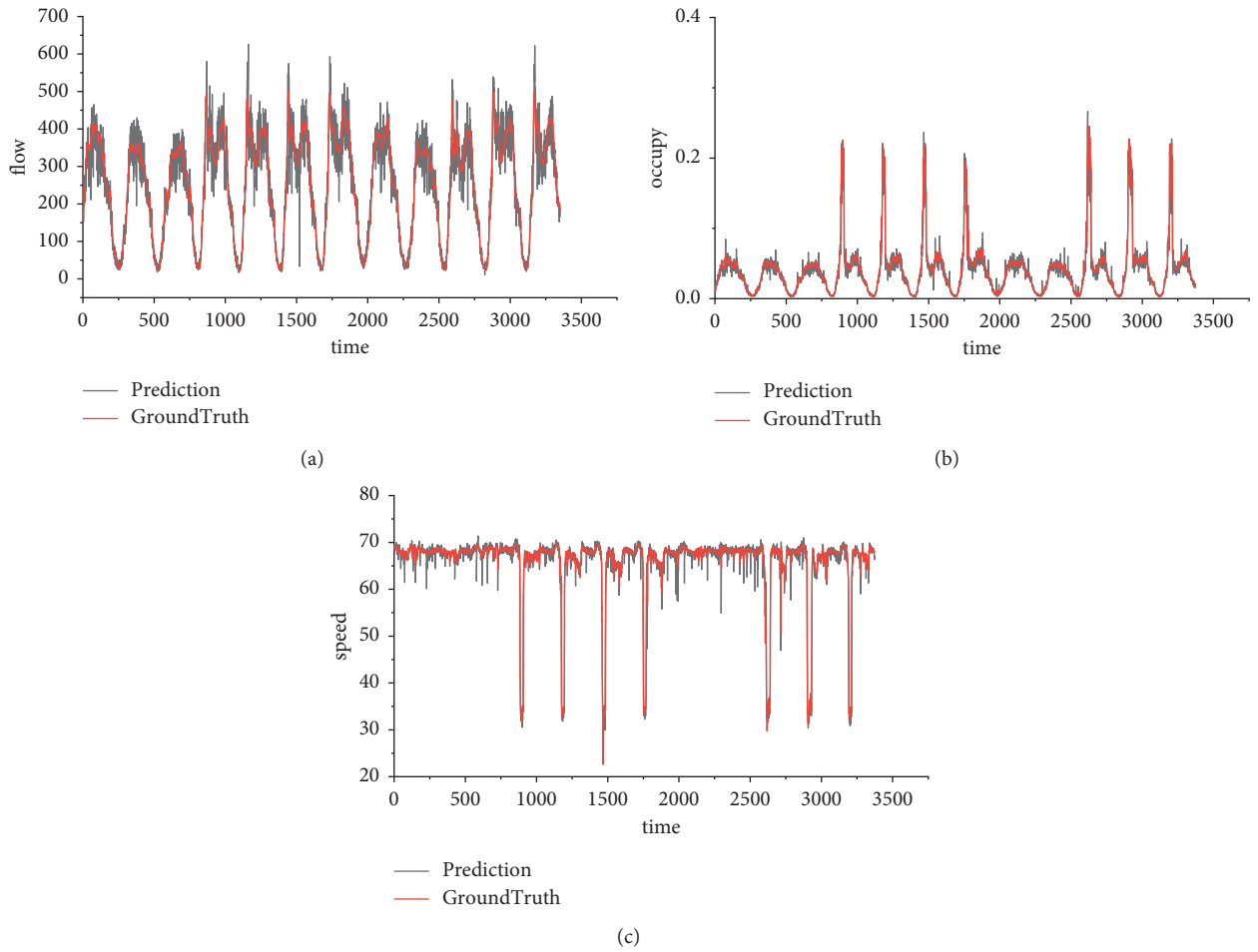

Figure 14: Comparison of the real values and predicted values for the three features on the PEMSD4 dataset. (a) Comparison of traffic flow. (b) Comparison of traffic occupy. (c) Comparison of traffic speed.

PEMSD8 dataset are 22.80, 0.01, and 1.62, while the MAPE metrics reach 9.49%, 9.91%, and 1.61%, respectively. Figures 16–18 show the prediction effects of the three characteristics at different horizons, respectively. With different horizon prediction experiments, the indicators all show an increasing trend with an increasing horizon. As shown in Figure 16, the range of RMSE metrics for flow on the PEMSD4 dataset increased from 31.62 to 35.35, while the range of RMSE on the PEMSD8 dataset also increased from 24.58 to 28.31. In Figure 17, the MAPE indicator of occupancy fluctuates widely, with a 31.63% increase in the predicted effect at 15 minutes versus 60 minutes. As can be observed in Figures 15–17, the indicators all show an increasing trend with the increasing horizon. We conjecture that because long-term forecasts do not contain as much information as short-term forecasts, AED-DGCN-TSC is able to learn enough useful information from the short-term historical data.

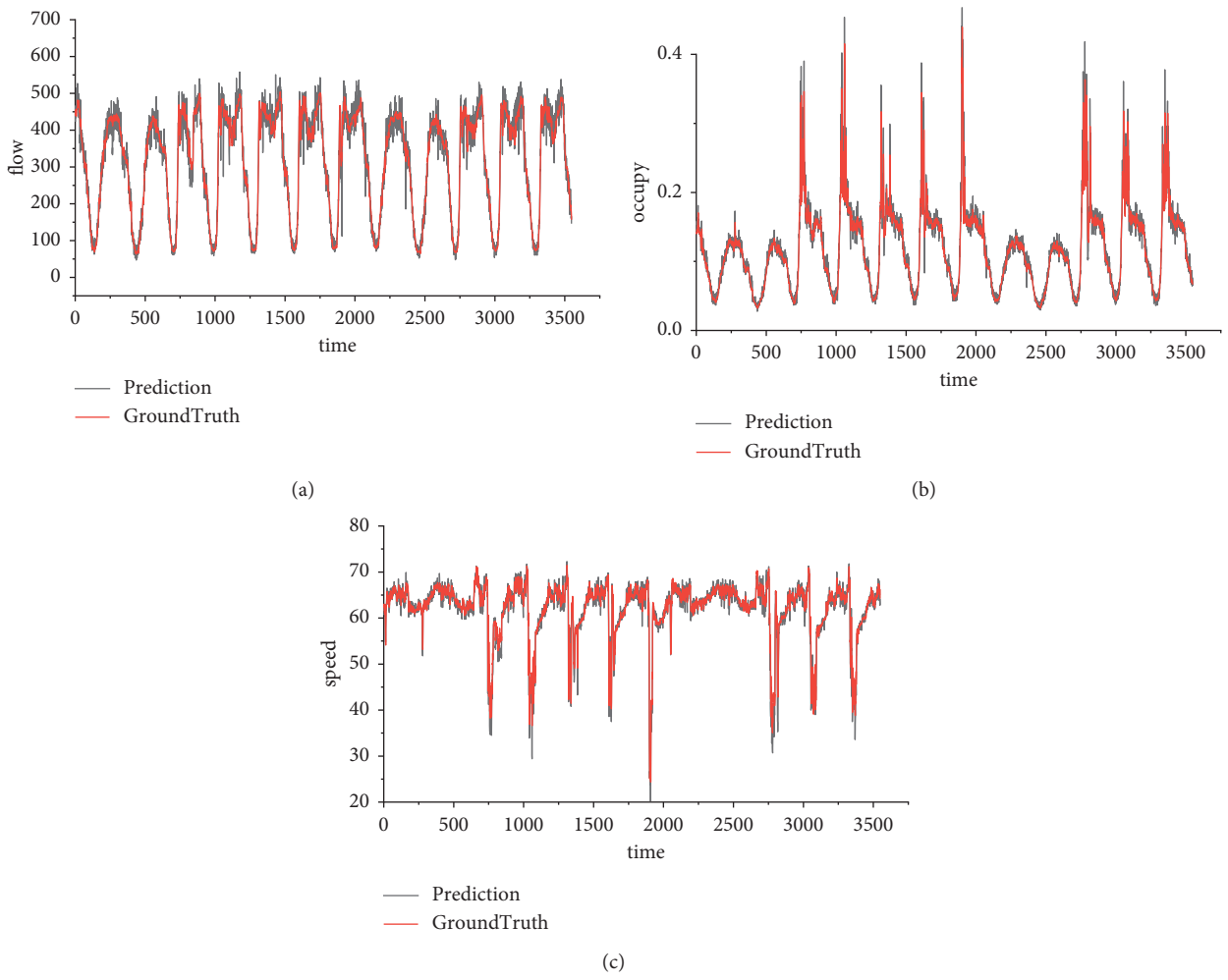

Figure 15: Comparison of the real values and predicted values for the three features on the PEMSD8 dataset. (a) Comparison of traffic flow. (b) Comparison of traffic occupancy. (c) Comparison of traffic speed.

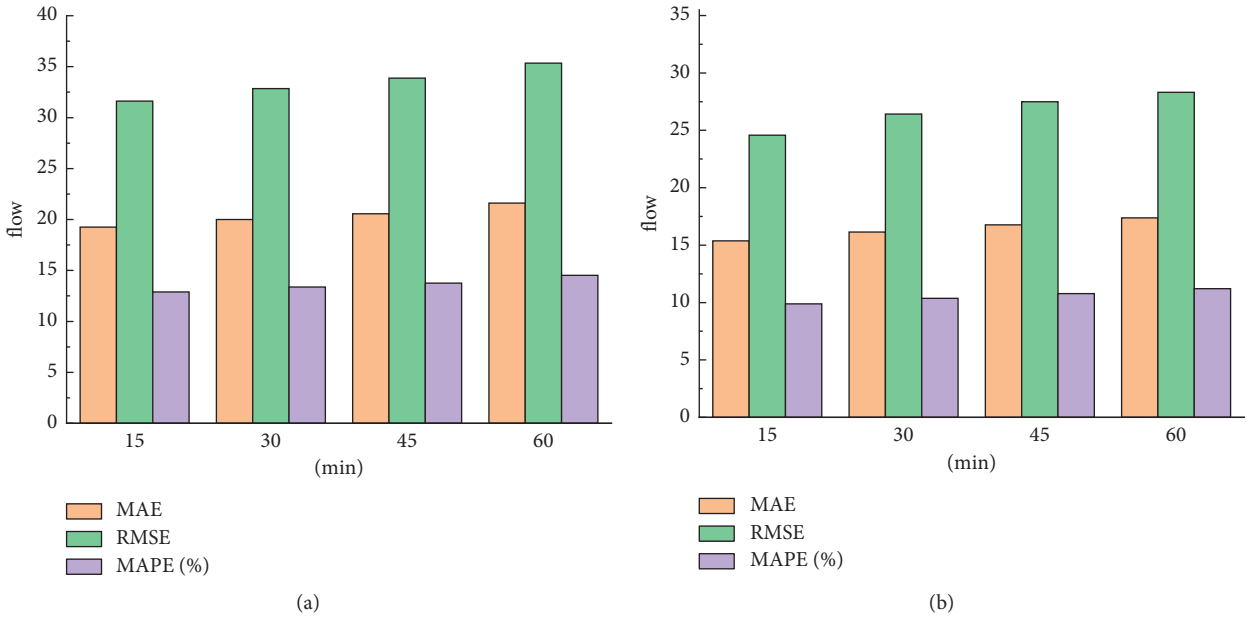

Figure 16: Performance comparison of flow on PEMSD4 and PEMSD8 datasets. (a) Performance comparison of flow on PEMSD4. (b) Performance comparison of flow on PEMSD8.

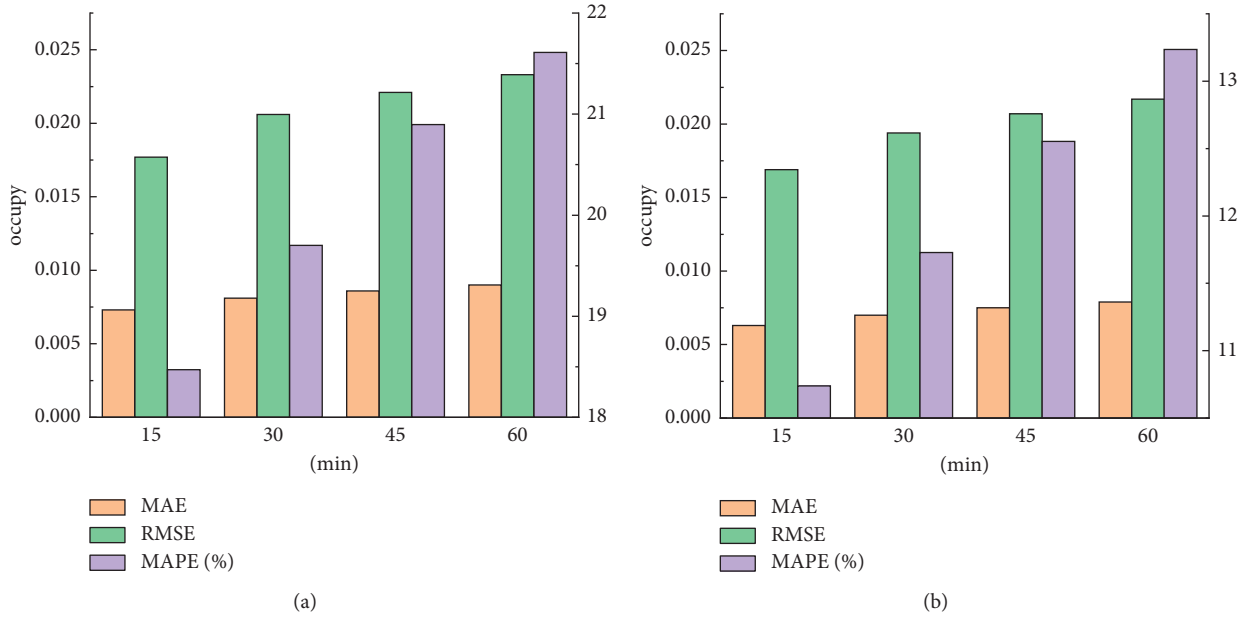

Figure 17: Performance comparison of occupy on PEMSD4 and PEMSD8 datasets. (a) Performance comparison of occupy on PEMSD4. (b) Performance comparison of occupy on PEMSD8.

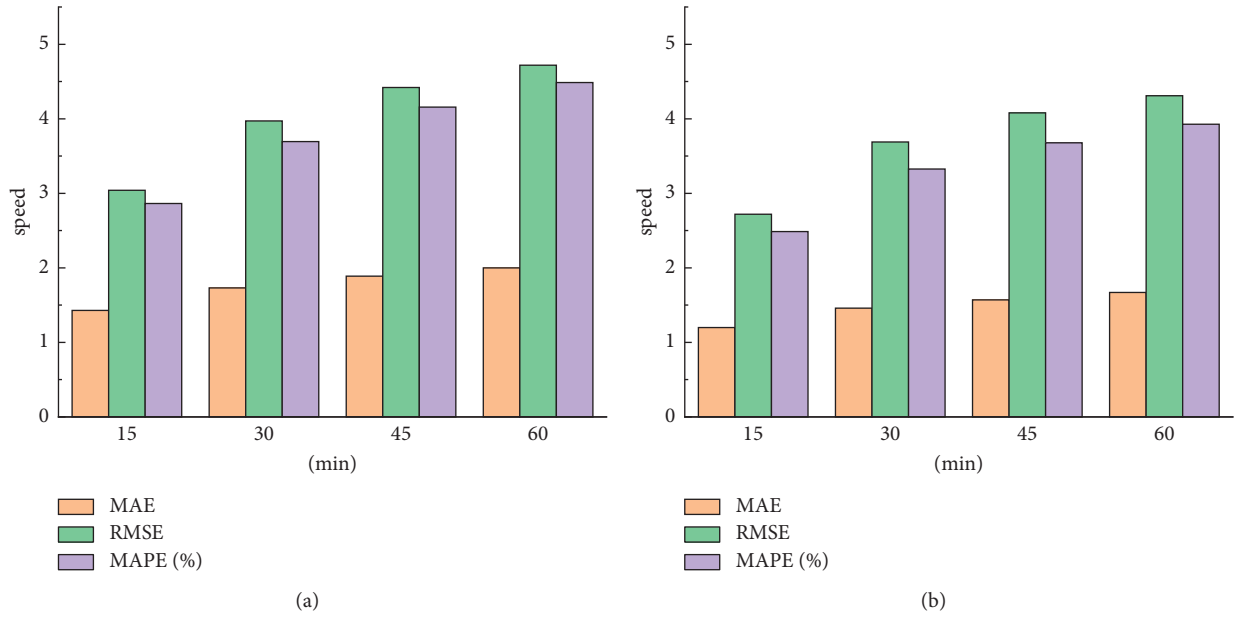

Figure 18: Performance comparison of speed on PEMSD4 and PEMSD8 datasets. (a) Performance comparison of speed on PEMSD4. (b) Performance comparison of speed on PEMSD8.

6. Conclusions

In this paper, we propose a new framework for predicting traffic flow characteristics based on the encoder-decoder structure. The proposed model can efficiently capture hidden spatial dependencies through a series of components. Representing the implicit graph node association information by means of embedding vectors, based on the concepts of pairwise graph convolution and gated loop structure, allows the model to learn both local and global spatio-temporal heterogeneity laws. Thanks to the idea of decomposition and similarity, the time-series correlation module performs aggregation operations on sequences within a certain delay period as similar subsequences and aligns them with the target sequence. Comparison with several benchmark models shows that AED-DGCN-TSC has advantages in capturing the spatio-temporal characteristics of traffic prediction. This work reveals the effective application of graph convolution in traffic flow sequence prediction by identifying dependencies from historical data,

while reflecting the importance of node learning for time-series prediction tasks. In addition, how to better fuse external features and multi-source data to enhance the prediction effect will be a future research work.

Data Availability

The dataset used in this paper is a public dataset constructed from a real collection and the access address is https://github.com/guoshnBJTU/ASTGCN-r-pytorch.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Innovation Fund Projects of Colleges and Universities in Gansu Province, China (no. 2021B-091).

References

[1] O.-P. Tossavainen and D. B. Work, "Markov chain Monte Carlo based inverse modeling of traffic flows using GPS data," *Networks and Heterogeneous Media*, vol. 8, no. 3, p. 803, 2013.

[2] 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.

[3] H. Sun, H. X. Liu, H. Xiao, R. H Rachel, and B. Ran, "Use of local linear regression model for short-term traffic forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1836, no. 1, pp. 143–150, 2003.

[4] W. Chen, J. An, R. Li et al., "A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features," *Future Generation Computer Systems*, vol. 89, pp. 78–88, 2018.

[5] L. Lin, Q. Wang, and A. W. Sadek, "A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 444–459, 2015.

[6] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *Journal of Advanced Transportation*, vol. 2019, pp. 1–10, Article ID 4145353, 2019.

[7] Z. Liu, W. Du, D. m. Yan, and C. Gan, "Short-term traffic flow forecasting based on combination of k-nearest neighbor and support vector regression," *Journal of Highway and Transportation Research and Development*, vol. 12, no. 1, pp. 89–96, 2018.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[9] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.

[10] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, pp. 7370–7377, Honolulu, HI, USA, January 2019.

[11] S. E. Kahou, X. Bouthillier, P. Lamblin et al., "EmoNets: multimodal deep learning approaches for emotion recognition in video," *Journal on Multimodal User Interfaces*, vol. 10, no. 2, pp. 99–111, 2015.

[12] P. Kumar and B. Bhasker, "DNNRec: a novel deep learning based hybrid recommender system," *Expert Systems with Applications*, vol. 144, Article ID 113054, 2020.

[13] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[14] X. Song, Y. Wu, and C. Zhang, "TSTNet: a sequence to sequence transformer network for spatial-temporal traffic prediction," in *Proceedings of the International Conference on Artificial Neural Networks*, pp. 343–354, Bratislava, Slovakia, September 2021.

[15] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 397–400, Seattle, WA, USA, November 2018.

[16] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 736–744, Marina Del Rey, CA, USA, February 2018.

[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, no. Nips, pp. 3844–3852, 2016.

[18] P. W. Battaglia, J. B. Hamrick, V. Bapst et al., "Relational Inductive Biases, Deep Learning, and Graph Networks," 2018, https://arxiv.org/abs/1806.01261.

[19] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful Are Graph Neural Networks?," 2018, https://arxiv.org/abs/1810.00826.

[20] S. Fu, W. Liu, D. Tao, Y Zhou, and L. Nie, "HesGCN: hessian graph convolutional networks for semi-supervised classification," *Information Sciences*, vol. 514, pp. 484–498, 2020.

[21] Y. Han, S. Wang, Y. Ren, and G. Chen, "Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks," *ISPRS International Journal of Geo-Information*, vol. 8, no. 6, p. 243, 2019.

[22] F. Hou, Y. Zhang, X. Fu, L. Jiao, and W. Zheng, "The prediction of multistep traffic flow based on AST-GCN-LSTM," *Journal of Advanced Transportation*, vol. 2021, Article ID 9513170, 10 pages, 2021.

[23] H. Zhu, Y. Xie, W. He et al., "A novel traffic flow forecasting method based on RNN-GCN and BRB," *Journal of Advanced Transportation*, vol. 2020, pp. 1–11, Article ID 7586154, 2020.

[24] J. Wang, W. Wang, X. Liu, W. Yu, X. Li, and P. Sun, "Traffic prediction based on auto spatiotemporal multi-graph adversarial neural network," *Physica A: Statistical Mechanics and Its Applications*, vol. 590, Article ID 126736, 2022.

[25] X. Chen, H. Chen, Y. Yang et al., "Traffic flow prediction by an ensemble framework with data denoising and deep learning model," *Physica A: Statistical Mechanics and Its Applications*, vol. 565, Article ID 125574, 2021.

[26] R. L. Abduljabbar, H. Dia, and P.-W. Tsai, "Unidirectional and bidirectional LSTM models for short-term traffic prediction," *Journal of Advanced Transportation*, vol. 2021, Article ID 5589075, 16 pages, 2021.

[27] W. Li, X. Wang, Y. Zhang, and Q. Wu, "Traffic flow prediction over muti-sensor data correlation with graph convolution network," *Neurocomputing*, vol. 427, pp. 50–63, 2021.

[28] S.-Y. Shih, F.-K. Sun, and H.-y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning*, vol. 108, no. 8-9, pp. 1421–1441, 2019.

[29] R. Zhang, F. Sun, Z. Song, X. Wang, Y. Du, and S. Dong, "Short-term traffic flow forecasting model based on GA-TCN," *Journal of Advanced Transportation*, vol. 2021, pp. 1–13, Article ID 1338607, 2021.

[30] X. Ma, H. Zhong, Y. Li, J Ma, Z. Cui, and Y. Wang, "Forecasting transportation network speed using deep capsule networks with nested LSTM models," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4813–4824, 2020.

[31] 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. 1, pp. 922–929, Honolulu, HI, USA, January 2019.

[32] 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.

[33] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph Wavenet for Deep Spatial-Temporal Graph Modeling," 2019, https://arxiv.org/abs/1906.00121.

[34] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 914–921, New York, NY, USA, February 2020.

[35] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, pp. 4189–4196, Palo Alto, CA, USA, february 2021.

[36] J. Xu, J. Wang, M. Long, and M. Long, "Autoformer: decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[37] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and H. Shi-Min, "Beyond Self-Attention: External Attention Using Two Linear Layers for Visual Tasks," 2021, https://arxiv.org/abs/2105.02358.

[38] F. Monti, O. Shchur, A. Bojchevski, O. Litany, S. Günnemann, and M. M. Bronstein, "Dual-primal Graph Convolutional Networks," 2018, https://arxiv.org/abs/1806.00770.

[39] 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.

[40] G. Yang, Y. Wang, H. Yu, Y. Ren, and J. Xie, "Short-term traffic state prediction based on the spatiotemporal features of critical road sections," *Sensors*, vol. 18, no. 7, p. 2287, 2018.

[41] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system - mining loop detector data," *Advanced Traffic Management Systems and Vehicle-Highway Automation*, vol. 1748, no. 1748, pp. 96–102, 2001.

[42] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, Long Beach, CA, USA, December 2017.

[43] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014, https://arxiv.org/abs/1412.3555.

[44] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting," 2018, https://arxiv.org/abs/1707.01926.

[45] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," 2017, https://arxiv.org/abs/1709.04875.

[46] F. Teng, J. Teng, L. Qiao, S. Du, and T. Li, "A multi-step forecasting model of online car-hailing demand," *Information Sciences*, vol. 587, pp. 572–586, 2022.

[47] J. Bao, H. Yu, and J. Wu, "Short term FFBS demand prediction with multi source data in a hybrid deep learning framework," *IET Intelligent Transport Systems*, vol. 13, pp. 1340–1347, 2019.