

Article

Seq2-ResGCRN: Sequence-to-Sequence Residual Graph Convolutional Recurrent Network for Traffic Flow Prediction

Wenyan Yan ^{1,2,*} and Tao Liu ³¹ School of Computer Science and Technology, Hunan First Normal University, Changsha 410205, China² Hunan Engineering Research Center of Advanced Embedded Computing and Intelligent Medical Systems, School of Computer and Artificial Intelligence, Xiangnan University, Chenzhou 423043, China³ Key Laboratory for Embedded and Network Computing of Hunan Province, College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China; ltliu@hnu.edu.cn

* Correspondence: ywy33@hnu.edu.cn

Abstract

Accurate traffic flow prediction remains challenging due to the complex spatio-temporal dependencies inherent in road networks. Existing graph-based prediction models typically construct the graph structure from geographical adjacency. This limits spatial feature extraction to physically neighboring nodes. Consequently, potential correlations among non-adjacent nodes within the same region are overlooked. Moreover, these models generally treat the influence of neighboring nodes uniformly, whereas, in practice, such influences are inherently heterogeneous. On the temporal side, many approaches rely on recurrent units such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRUs) to capture short-term dependencies, yet these architectures are prone to gradient vanishing or explosion, leading to training instability. To address these issues, this paper proposes Seq2-ResGCRN, an encoder–decoder framework based on residual graph convolutional recurrent networks with an attention mechanism. The model reconstructs the adjacency matrix via graph diffusion convolution to learn adaptive edge weights, thereby capturing heterogeneous spatial influences. Seq2-ResGCRN further integrates graph convolution into the GRU architecture and introduces residual connections to alleviate gradient degradation during training. This study conducts experiments on two open-source real-world datasets (i.e., PEMS04 and PEMS08) to evaluate the Seq2-ResGCRN. The raw data, originally collected at a sampling frequency of 30 s, are aggregated into 5 min time intervals. Each record comprises three features: traffic flow, traffic speed, and road occupancy. The experimental results demonstrate that our Seq2-ResGCRN outperforms state-of-the-art methods, achieving 1.6–2.3% and 2.1–5.5% relative improvements in MAE and RMSE, respectively. Seq2-ResGCRN effectively captures the spatio-temporal correlations of short-term traffic flow and achieves superior predictive performance.



Academic Editor: Nadir Farhi

Received: 30 July 2026

Revised: 11 September 2026

Accepted: 15 September 2026

Published: 17 September 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: traffic flow prediction; deep learning; temporal and spatial correlation; graph convolution; attention

MSC: 68T07; 90B20

1. Introduction

Rapid urbanization has attracted an increasing number of people to cities. Substantial vehicle ownership and travel demand have placed tremendous pressure on urban public transportation management [1]. Traffic congestion and traffic accidents pose significant

threats to public mobility [2,3]. Addressing these issues to satisfy daily travel needs has become an unavoidable task. Intelligent transportation systems (ITSs) offer new insights into addressing traffic problems [4]. Accurate road traffic flow prediction plays a critical role in the implementation of such systems [5–7]. In an era when traffic data is scarce and computer hardware is limited, road traffic flow prediction is primarily modeled using statistical methods and conventional machine learning algorithms [8]. With the continuous development of the Internet of Things (IoT) technology, a large volume of rich road traffic data can now be collected through sensors [9]. However, statistical methods and traditional machine learning approaches suffer from issues such as low accuracy and modeling difficulty when applied to large-scale traffic data. In recent years, deep learning-based algorithms have become a research focus in the field of natural language processing and have also demonstrated great potential in road traffic flow prediction [10].

Short-term road traffic flow is characterized by nonlinear temporal dynamics and complex spatiotemporal dependencies [11–13]. This is reflected in the marked differences in traffic flow between peak and off-peak hours, as well as the mutual influence among adjacent road nodes. Additionally, short-term traffic flow is also affected by sudden events, including traffic accidents, temporary road closures, and abrupt weather changes [14–16]. The combined effect of these multiple factors makes short-term traffic flow prediction particularly challenging. Early researchers primarily combined Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for short-term road traffic flow prediction, leveraging CNNs to extract spatial features and RNNs to capture temporal characteristics of the traffic flow [17]. However, a substantial body of research has demonstrated that CNNs perform suboptimally in extracting features from irregular network topologies [18]. The development of graph neural networks has provided new insights into extracting spatial features from irregular network structures.

Numerous studies have compared GCN-based hybrid models against CNN-RNN baselines. The results show that GCNs combined with RNNs or TCNs (Temporal Convolutional Network) achieve superior performance. This consistently confirms the feasibility of applying graph convolution to traffic flow prediction. However, some studies directly apply graph convolution to the original graph structure data of the road network, which is derived from the geographic neighbor relationships among observation nodes. In this process, each node updates its representation by aggregating feature information exclusively from its geographically neighboring nodes. In real-world road networks, traffic flow between two non-adjacent nodes also exhibits mutual influences. For example, within the same region, if a destination is reachable via two alternative routes, traffic congestion on one route may prompt travelers to divert to the other, leading to a temporary surge in traffic flow on that route. However, the nodes of these two routes are not geographically adjacent. Therefore, the original graph structure derived from geographic proximity cannot capture such relationships, and also fails to reflect the varying influence that different neighboring nodes exert on the current node's traffic flow.

Recent studies have attempted to address the above limitation through adaptive graph learning. For instance, Wu et al. [19] proposed constructing an adaptive adjacency matrix to infer latent dependencies from data; Bai et al. [20] combined graph convolution with node-adaptive parameter learning and data-adaptive graph generation; and Shin et al. [21] introduced a progressive graph convolution network that learns similarity in traffic variation trends to construct a progressive adjacency matrix. Li et al. [22] employed Dynamic Time Warping to measure traffic sequence similarity and construct time-series-based graph structures; Rao et al. [23] generated spatio-temporal correlation graphs based on average traffic flow; and Guo et al. [24] partitioned road network graphs into local and global sub-graphs. While these methods have demonstrated promising performance, they typically

integrate graph convolution with recurrent units in a sequential or parallel manner, without modifying the recurrent unit's internal structure to explicitly mitigate gradient-related instability during training.

Existing studies rely on original Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) architectures to extract short-term temporal features of road traffic flow [25–28]. During the training process, however, the gradient vanishing or explosion problem inherent in GRU can lead to model instability, thereby hindering the effective extraction of temporal features. To address the aforementioned issues, this chapter proposes a novel short-term road traffic flow prediction model, termed the Residual Graph Convolutional Recurrent Network (ResGCRN), to capture the short-term evolution patterns of road traffic flow. The proposed model differs from existing approaches in three key aspects. First, we construct an adaptive adjacency matrix via graph diffusion convolution to capture both geographic and non-geographic dependencies, refined during training. Unlike prior adaptive graph methods [19–24] that fuse adaptive matrices sequentially, our approach directly integrates the learned graph into the recurrent unit's convolution, enabling tighter coupling of spatial structure learning and temporal evolution. Second, we embed graph convolution and residual connections into the GRU cell to mitigate gradient vanishing and explosion (a design not explored in existing adaptive-graph-based models). Third, we equip the encoder–decoder with ResGCRN modules and attention to strengthen spatio-temporal dependency modeling. We have made the following contributions:

- We propose the Seq2-ResGCRN network model, which constructs an adaptive adjacency matrix and applies graph diffusion convolution to derive a learnable graph structure, while incorporating graph convolution and residual connections into the GRU cell to mitigate gradient vanishing and explosion.
- We build an encoder–decoder architecture with ResGCRN modules and integrate an attention mechanism to enhance spatio-temporal dependency modeling.
- We compare the proposed model with several representative traffic flow prediction algorithms. The experimental results demonstrate that Seq2-ResGCRN can effectively capture the spatio-temporal dependencies of short-term traffic flow.

2. Problem Formulation

The task of road traffic flow prediction is to process and analyze historical traffic flow data through algorithmic models, extract the underlying evolution patterns of traffic flow, and subsequently forecast future traffic conditions over a given time horizon. This task can be viewed as a regression problem, where prediction accuracy improves as the discrepancy between the predicted values and the ground-truth observations decreases.

Unlike conventional regression tasks, traffic flow exhibits complex temporal and spatial variations. It is influenced not only by temporal factors but also by road network topology, weather conditions, holidays, traffic incidents, and other external covariates. However, due to the frequent incompleteness of such auxiliary data, only a limited set of influencing factors can typically be incorporated in practice. In this study, we therefore restrict our inputs to historical traffic flow data, focusing on modeling the inherent spatio-temporal dependencies without incorporating external covariates.

In the context of deep learning-based approaches for traffic flow prediction, when considering solely the spatio-temporal correlations of traffic flow, the underlying road network is commonly represented as a directed graph, as is shown in Equation (1).

$$G = (V, E, A), \quad (1)$$

where V denotes the set of nodes corresponding to N traffic sensors; E denotes the set of edges representing connectivity among sensors; A is the adjacency matrix that encodes the graph topology.

$X = \{x_{it}\} \in \mathbb{R}^{N \times T}$ denotes the traffic flow matrix, where T is the number of time steps. The objective of traffic flow prediction is to learn a mapping function f . f forecasts the traffic flow for the next h time steps given the observations from the previous k time steps. This relationship can be formulated as

$$\{x_t, x_{t+1}, \dots, x_{t+h-1}\} = f(x_{t-1}, x_{t-2}, \dots, x_{t-k}, G), \quad (2)$$

where $x_{t-k:t-1}$ represents the historical traffic sequence of length k , and $x_{t:t+h-1}$ denotes the predicted sequence for the future h time steps.

3. Short-Term Traffic Flow Feature Analysis

Traffic flow prediction can be categorized into short-term and long-term forecasting according to the temporal scale. Short-term prediction typically refers to forecasting horizons ranging from several minutes to a few hours, focusing on capturing real-time traffic flow fluctuations. In contrast, long-term prediction addresses macroscopic trends over days to years, supporting urban planning and infrastructure decision-making. The scope of this study is confined to short-term traffic flow prediction.

Short-term road traffic flow primarily concerns traffic variations within the next half hour. Given the short prediction horizon, the short-term flow changes at a given observation point are largely dependent on its traffic evolution trend over the most recent hour. In addition, the low sampling rate of the data contributes to the complex and volatile nature of short-term traffic flow. The mutual influences among different observation points across the road network further compound the difficulty of prediction. Figure 1 illustrates the variation trend of short-term road traffic flow.

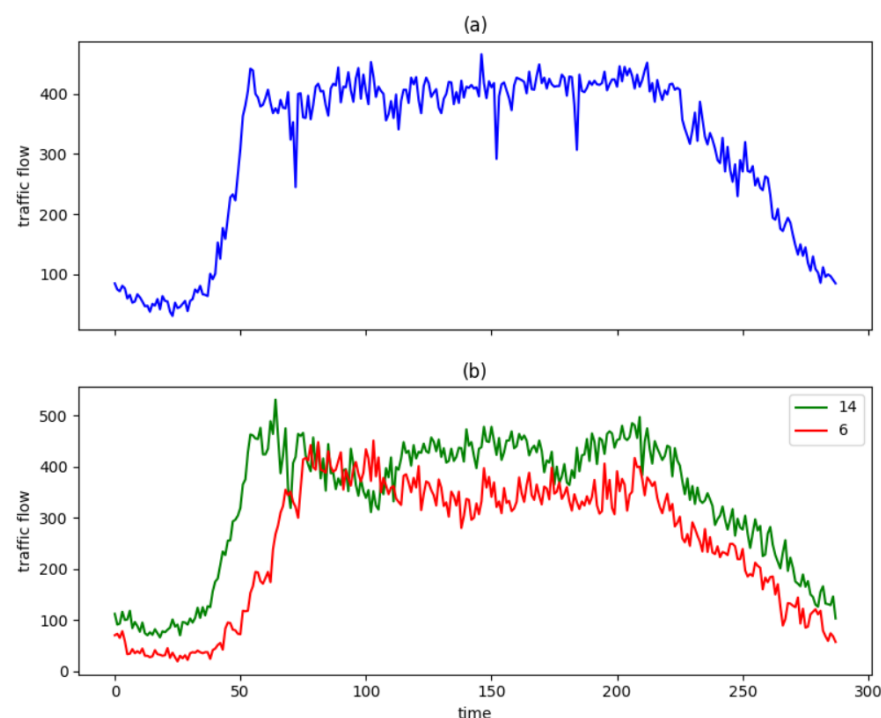


Figure 1. Variation trend of short-term road traffic flow. (a) Short-term traffic flow variation trend at a specific node in the road network; (b) variation trends of traffic flow at two adjacent observation points in the road network.

Figure 1a depicts the short-term traffic flow variation trend at a specific node in the road network over the course of a day. We can observe that the traffic flow at this node changes at every time step, highlighting the highly dynamic and non-stationary nature of short-term traffic data. Within a short time window, traffic flow may experience abrupt surges or drops, exhibiting highly complex nonlinear variations. Figure 1b depicts the variation trends of traffic flow at two adjacent observation points in the road network. Although the traffic volumes differ in magnitude, the overall variation patterns at the two points are roughly similar. Meanwhile, the traffic flow at observation node 14 propagates to node 6 after a certain time delay, indicating that traffic flows between adjacent nodes in the road network mutually influence each other. Consequently, the current predicted flow at a given node is the result of both its own historical traffic data and the historical flow data of its neighboring observation nodes.

4. Model Architecture of Seq2-ResGCRN

This section describes the specific process of short-term road traffic flow prediction using the Seq2-ResGCRN network. The Seq2-ResGCRN model consists of three main components: an encoder network, a decoder network, and an attention computation module. The specific process are as follows:

- (1) The road network information and historical traffic flow data are fed into the encoder network to obtain the hidden state outputs of the encoder.
- (2) The hidden state output at the last time step of the encoder is used as the initial hidden state for the decoder. During the neural network's information updating process, the hidden state outputs at each time step of the decoder are obtained through iterative computation. The attention weight coefficients are then computed using the decoder's hidden state outputs at each time step and the encoder's hidden state outputs. The current attention weights are then concatenated with the decoder output at the current time step. The resulting vector is processed through a fully connected layer (FC) to produce the current prediction. This prediction is subsequently fed back into the decoder as the input for the next time step.
- (3) Through this iterative prediction process, the final forecasting results are obtained. The overall framework of the Seq2-ResGCRN model is illustrated in Figure 2.

4.1. Graph Convolution Module

Graph convolution theory extends the application of convolution from regular grid-structured data to irregular topological graph structures. In the context of road traffic flow prediction theory, the spatial structure of the road network can be naturally regarded as an irregular spatial graph, where the traffic flow at each observation point can be viewed as the signal of the corresponding node at a given time. The connectivity of the graph is represented by an adjacency matrix, where the entry is set to 1 if two nodes are adjacent, and 0 otherwise.

$$A_{i,j} = \begin{cases} 1, & \text{if } v_i \text{ connects to } v_j \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where v_i and v_j denote nodes in the spatial structure of the road network; N represents the total number of nodes in the network. In the field of graph neural networks, when the original topological graph contains noisy connections, the performance of graph neural networks may deteriorate, further affecting downstream learning tasks. On the other hand, in real-world scenarios, the influence of different neighboring nodes on the current node should be differentiated, a relationship that the geographically defined graph structure

fails to capture. To address this issue, we introduce the graph diffusion convolution (GDC) operation in the design of the graph convolution module to optimize the adaptive adjacency matrix, thereby enabling more effective learning of the graph structure.

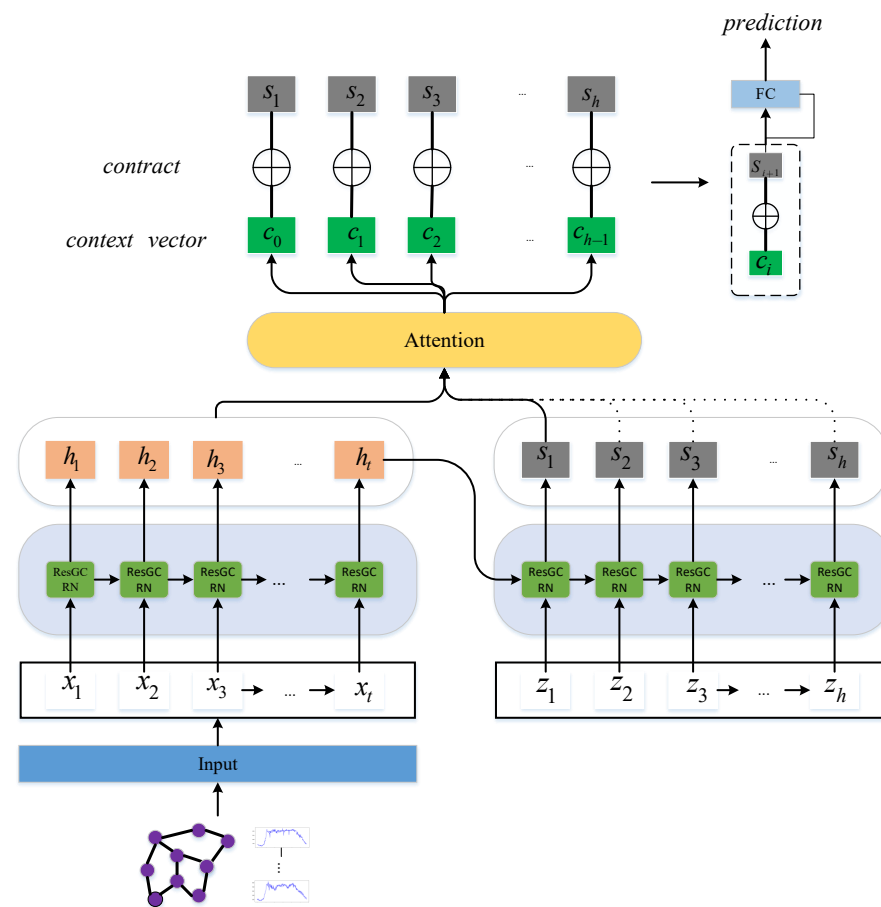


Figure 2. Network structure of Seq2-ResGCRN.

(1) Adaptive Graph Generation Module

The road network topology in the original dataset captures the spatial adjacency relationships among traffic flow collection points. In real-world road networks, nodes are not only connected by direct adjacency but also show indirect correlations. For instance, when a node experiences congestion, travelers may reroute to the same destination through other non-adjacent nodes in the region. This indicates non-adjacent nodes still affect each other. However, the adjacency matrix of the original road spatial topology fails to capture such indirect correlations. Therefore, it is necessary to adopt the learned road network graph structure for prediction. In this paper, we adopt an adaptive adjacency matrix to replace the road network information provided in the dataset. This matrix is built upon two randomly initialized learnable parameter sets for node encoding.

The learnable parameter $E_1, E_2 \in \mathbb{R}^{N \times C}$ denotes node information, where E_1 and E_2 are the encoding matrices of the nodes; N is the number of road network nodes; and C is the dimension of node embeddings. The product of E_1 and $E_2^\top$ is then computed to obtain the dependency weights between source and target nodes. A *ReLU* activation function is applied to eliminate weak connections. Finally, a softmax function is used to normalize the adaptive adjacency matrix, ensuring that its values lie between 0 and 1. This process can be expressed as

$$\tilde{A}_{adp} = \text{softmax}\left(\text{ReLU}\left(E_1 E_2^\top\right)\right), \quad (4)$$

where $\tilde{A}_{adp} \in \mathbb{R}^{N \times N}$ denotes the adaptive adjacency matrix.

(2) Graph Diffusion Convolution Module

Graph convolutional networks are widely used for feature extraction on non-Euclidean graphs. In road networks, neighboring nodes affect the target node's traffic flow to varying degrees. For example, at a signalized intersection, through and turning traffic volumes often differ substantially. However, the predefined adjacency matrix derived from the raw road network topology only records connectivity and distance between nodes, without incorporating edge weights. That is, the static graph lacks the capability to differentiate the influence of distinct neighboring nodes. To address this limitation, we introduce an adaptive adjacency matrix and incorporate graph diffusion convolution to optimize the graph structure, thereby better capturing the spatial influence relationships of traffic flow among different neighboring nodes in the road network.

To solve the above problem, we adopt graph diffusion convolution [29] to improve the adaptive adjacency matrix. The key idea is to aggregate information from both first-hop and higher-order neighbors. Initially, all attention is concentrated on the current node. During graph diffusion, attention is progressively propagated to neighboring nodes and weighted differently based on distance. Nearby nodes receive higher weights, while distant nodes receive lower weights. During graph reconstruction, edges with small weights are removed. This process is repeated for all nodes, ultimately yielding a sparse weighted graph. Figure 3 illustrates the workflow of graph diffusion convolution.

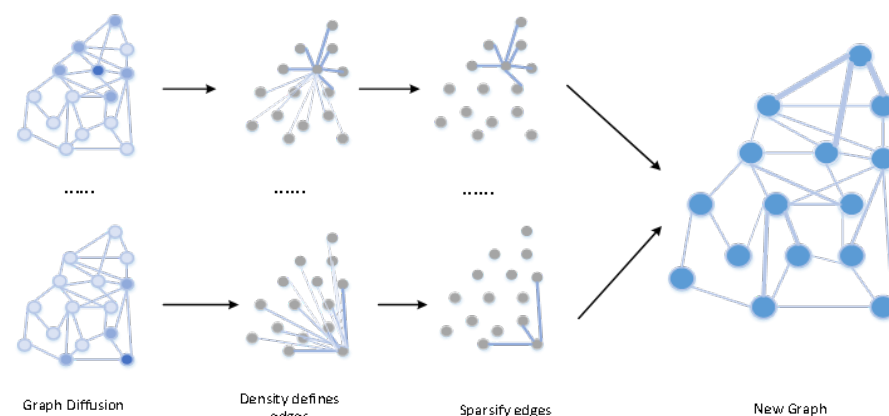


Figure 3. Workflow of graph diffusion convolution.

As shown in Figure 3, the essence of graph diffusion convolution is to transform the input graph A into a new graph $\tilde{S}$ through graph diffusion and sparsification, and then run a given model on the new graph. For a directed graph $G = (V, E)$ with node set V and edge set E , its adjacency matrix is $A \in \mathbb{R}^{N \times N}$, where N is the number of nodes in the graph. A generalized graph diffusion can be defined via the diffusion matrix, expressed as

$$S = \sum_{k=0}^{\infty} \theta_k T^k, \quad (5)$$

where θ_k is the weighting coefficient and T is the state transition matrix. To ensure the convergence of Equation (5), the following conditions must be satisfied

$$\sum_{k=0}^{\infty} \theta_k = 1, \quad \theta_k \in [0, 1], \quad \lambda_i \in [0, 1], \quad (6)$$

where λ_i denotes the eigenvalues of T . The state transition matrix is obtained by incorporating self-loops into the original adjacency matrix to adjust the symmetric transition matrix, given by

$$T = (w_{\text{loop}}I_N + D)^{-\frac{1}{2}}(w_{\text{loop}}I_N + A)(w_{\text{loop}}I_N + D)^{-\frac{1}{2}} \quad (7)$$

Furthermore, for graph diffusion, a PPR (PageRank) diffusion method is proposed in [30]. The weighting coefficients are defined as:

$$\theta_k = \alpha(1 - \alpha)^k, \quad (8)$$

where $\alpha \in (0, 1)$. By substituting Equations (7) and (8) into Equation (5), the graph diffusion matrix S is obtained. A threshold is then applied to retain only the entries in S that exceed the threshold, yielding a sparse graph $\tilde{S}$. Finally, the graph convolution process is defined as:

$$GC(x, \tilde{S}) = (D^{\frac{1}{2}}\tilde{S}D^{-\frac{1}{2}})x, \quad (9)$$

where D represents the degree matrix.

4.2. ResGCRN Network

Road traffic flow is typical spatio-temporal data with complex temporal and spatial dependencies. In this section, we propose the Residual Graph Convolutional Recurrent Network (ResGCRN) to simultaneously capture these spatio-temporal dependencies. ResGCRN improves the conventional GRU unit by incorporating graph convolution and residual connections within the GRU architecture. This design not only captures both spatial and temporal dependencies simultaneously, but also mitigates the vanishing gradient problem that may arise during GRU training. The architecture of ResGCRN is shown in Figure 4:

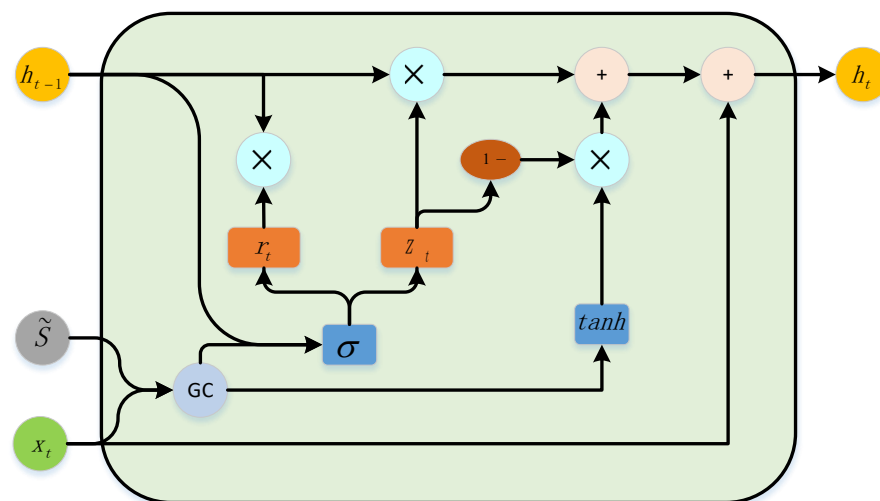


Figure 4. Architecture of ResGCRN.

As shown in Figure 4, compared with the standard GRU, ResGCRN introduces an additional graph input parameter to enable graph convolution operations. ResGCRN takes three inputs: the hidden state h_{t-1} from the previous time step, the current input x_t , and the road network topology $\tilde{S}$. The update process for each gating mechanism is reformulated as follows:

$$r_t = \sigma(W_r * h_{t-1} + GC(x_t, \tilde{S}) + b_r), \quad (10)$$

$$z_t = \sigma(W_z * h_{t-1} + GC(X_t, \tilde{S}) + b_z), \quad (11)$$

$$\tilde{h}_t = \tanh(GC(X_t, \tilde{S}) + W_c * (r_t \odot h_{t-1}) + b_c), \quad (12)$$

$$h_t = (1 - z) \odot h_{t-1} + z_t \odot \tilde{h}_t + W * X_t, \quad (13)$$

where W_r , W_z , W_c , and W are learnable training parameters, GC denotes the graph convolution operation, X_t is the input at time t , and $\tilde{S}$ is the sparse graph structure generated by applying graph diffusion convolution to the adaptive adjacency matrix.

4.3. Encoder–Decoder Architecture

The Seq2Seq architecture combined with an attention mechanism was first applied to machine translation models [31]. In this section, we design an encoder–decoder network model based on the ResGCRN architecture denoted as Seq2-ResGCRN. The encoder maps the input sequence of variable length into a fixed-length vector representation. The decoder then transforms this fixed-length vector into an output sequence of variable length. The Seq2Seq architecture overcomes the limitations of fixed-size input and output frameworks. The Seq2Seq can process sequences of variable length, offering greater flexibility. Meanwhile, the introduction of the attention mechanism can further improve the predictive performance of the model.

The encoder processes the traffic flow sequences across t historical time steps and generates the corresponding hidden state information. This process can be formulated as

$$[h_1, h_2, \dots, h_t] = \text{ResGCRN}(\tilde{S}, x_1, x_2, \dots, x_t).$$

The decoder takes the traffic flow input at the last time step of the encoder as its initial input, and adopts the hidden state output at the last time step of the encoder as its initial hidden state. Meanwhile, the input information to the decoder is processed by ResGCRN to generate the corresponding hidden state information of the decoder, which can be formulated as

$$[s_1, s_2, \dots, s_h] = \text{ResGCRN}(\tilde{S}, z_1, z_2, \dots, z_h). \quad (14)$$

Let H denote the hidden state information generated by the encoder. The hidden state output at each decoding step needs to compute attention weights with H . This computation process consists of three steps: linear transformation, dot product, and normalization. The first step is to generate the Key and Query through linear transformation

$$k_i = W_k \cdot h_i, \quad (15)$$

$$q_j = W_Q \cdot s_j, \quad (16)$$

where W_k and W_Q are learnable parameter matrices, with $i \in (1, t)$ and $j \in (0, h)$. Subsequently, the dot product is computed to obtain the candidate attention weights $\tilde{a}_i$. Finally, the softmax function is applied to normalize these candidate weights, yielding the final attention coefficients. The context vector is then obtained by computing the weighted sum of the attention coefficients and the hidden state outputs of the encoder at each time step

$$\tilde{a}_i = k_i^T \cdot q_i, \quad (17)$$

$$[a_1, a_2, \dots, a_m] = \text{softmax}[\tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_m], \quad (18)$$

$$c_j = \sum_{i=1}^t a_i \cdot h_i. \quad (19)$$

For the hidden state output of the decoder, the computation process after incorporating the attention mechanism is expressed as

$$s_j = \tanh \left(\text{ResGCRN} \begin{bmatrix} z_j \\ s_{j-1} \\ c_{j-1} \end{bmatrix} + b \right). \quad (20)$$

The initial input to the decoder is set as the input at the last time step of the encoder, i.e., $z_1 = x_t$. Subsequently, the prediction result from the previous time step is used as the input at the next time step, which can be formulated as:

$$z_j = FC(s_{j-1} \oplus c_{j-2}), \quad (21)$$

where $j \in (2, h)$, and through iterative prediction over multiple time steps, the final prediction result is obtained.

5. Experimental Evaluation and Analysis

5.1. Experimental Setup

In this section, all experiments are conducted on the same server. The hardware specifications are as follows: Intel Core i7-10700K CPU at 3.80 GHz, 64 GB RAM, 1 TB HDD, and an NVIDIA GeForce RTX 3080 GPU with 10 GB of VRAM.

The operating system used for the experiments was Windows 10 Professional. The code is written and debugged using PyCharm 20.1. The experimental runtime environment is a Python development environment created by Anaconda, with Python version 3.8. In addition, since deep learning is employed for model development, PyTorch 1.9.0 is used to build the deep learning models. To fully utilize GPU resources, CUDA is used to accelerate the training process on the GPU.

5.2. Dataset Description

To evaluate the proposed method, we conduct experiments on two publicly available real-world traffic datasets (PEMS04 and PEMS08) provided by the California Department of Transportation (Caltrans) [32]. The raw data (originally recorded at 30 s sampling intervals) is aggregated into 5 min time intervals. Each record contains three features: traffic flow, average speed, and lane occupancy. The PEMS04 dataset covers traffic data from the San Francisco Bay Area from 1 January 2018 to 28 February 2018, with 307 sensor stations. The PEMS08 dataset covers the San Bernardino area from 1 July 2016 to 31 August 2016, with 170 sensor stations. Detailed statistics of the datasets are summarized in Table 1.

Table 1. PEMS04 and PEMS08 datasets.

Dataset	Days	Sensors	Time Interval	Time Steps	Time Range
PEMS04	59	307	5 min	16,992	1 January 2018–28 February 2018
PEMS08	62	170	5 min	17,856	1 July 2016–31 August 2016

We select traffic flow as the prediction target from the dataset. It is worth noting that we use only historical traffic flow data as model inputs. A sliding window approach is applied to preprocess the raw data, generating the corresponding data and label. Specifically, data represents the historical window sequence, while label corresponds to the prediction window sequence. The dataset is split into training, test, and validation sets with a ratio of 7:1:2. To facilitate data processing and model training, the flow data are standardized as follows.

$$X' = \frac{X - \text{mean}(\text{train})}{\text{std}(\text{train})}, \quad (22)$$

where $\text{mean}(\text{train})$ and $\text{std}(\text{train})$ denote the mean and standard deviation of the training set, respectively.

5.3. Baseline Methods

To evaluate the effectiveness of the proposed Seq2-ResGCRN model, we compare Seq2-ResGCRN with eight classical road traffic flow prediction algorithms, including HA, VAR [33], SVR [34], LSTM [27], GRU [28], STGCN [35], TCN [36], and ASTGCN [37]. These algorithms encompass a wide range of representative methods, spanning traditional statistical approaches, classical machine learning models, and the most prominent deep learning architectures. All of these baselines are widely recognized and extensively employed in the domain of traffic flow prediction. The baseline methods are briefly introduced as follows:

HA: The historical average (HA) algorithm predicts the traffic flow by computing the average of historical data. In our experiments, the traffic flow from the past one hour is used as the historical data to obtain the prediction results.

VAR [33]: The vector autoregression algorithm extends the autoregression (AR) algorithm to enable predictions on high-dimensional data.

SVR [34]: Support vector regression employs a linear support vector machine for regression analysis tasks.

LSTM [27]: A variant of RNN that incorporates gating units to mitigate the gradient vanishing problem in RNN training.

GRU [28]: A variant of the LSTM neural network that significantly reduces the number of training parameters.

STGCN [35]: The spatio-temporal graph convolutional network employs ChebNet and 2D convolutions to capture the spatial and temporal dependencies of traffic flow, respectively.

TCN [36]: The temporal convolutional network is a time series prediction model that utilizes causal convolutions and dilated convolutions.

ASTGCN [37]: The attention-based spatio-temporal graph convolutional network models spatio-temporal correlations through spatial and temporal attention mechanisms. For a fair comparison, only the recent component is used to model the periodicity of traffic flow data.

5.4. Experimental Results

The mean absolute error (MAE) and root mean square error (RMSE) are adopted as evaluation metrics. MAE is the arithmetic mean of the absolute differences between the predicted and true values. A larger MAE indicates a larger prediction error of the model:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i^{\text{pred}} - y_i^{\text{true}}|. \quad (23)$$

RMSE measures the deviation between the predicted and true values. A smaller RMSE indicates better predictive performance

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i^{\text{pred}} - y_i^{\text{true}})^2}. \quad (24)$$

Figure 5 illustrates the loss variation in the Seq2-ResGCRN model on the PEMS04 dataset with a prediction horizon of 6. For brevity, we present the training loss curves for a representative prediction horizon (i.e., horizon = 6) in Figure 5. After four epochs, both the

training loss and validation loss gradually level off, and the training eventually converges and terminates at around 30 epochs under the early stopping mechanism.

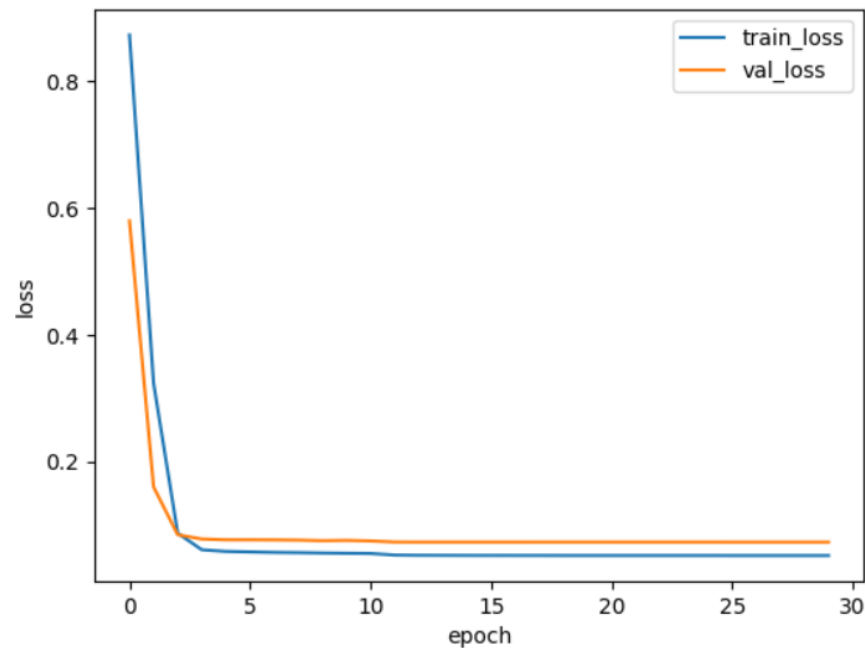


Figure 5. Training loss curves of Seq2-ResGCRN.

For all experiments, we use the traffic flow from the past one hour to predict the traffic flow for the next one hour. Specifically, 12 historical time steps are used to predict the next 12 time steps. The MAE and RMSE values are computed for each time step, summed, and then averaged across all steps. For the experimental data, we split the training set, validation set, and test set in a ratio of 7:1:2. By default, the model is trained for 50 epochs with an initial learning rate of 0.001. To prevent overfitting, we adopt the Adam optimizer for training and manually reduce the learning rate at every multiple of 10 epochs. If the validation loss does not decrease for seven consecutive epochs, early stopping is triggered to terminate training prematurely, thereby achieving the optimal training performance. All experimental results are obtained from the equipment described in Section 5.1. The final results are summarized in Table 2.

Table 2. Experimental results of various models on the PEMS04 and PEMS08 datasets.

Method	PEMS04		PEMS08	
	MAE	RMSE	MAE	RMSE
HA	32.10	46.72	26.51	39.36
VAR [33]	25.19	38.60	19.79	29.29
SVR [34]	23.39	36.28	18.59	28.74
LSTM [27]	25.41	39.32	20.29	30.21
GRU [28]	24.72	38.37	19.49	28.59
STGCN [35]	21.94	34.78	18.06	28.09
TCN [36]	26.07	40.22	20.67	30.98
ASTGCN [37]	22.16	35.19	18.10	28.15
Seq2-ResGCRN (Ours)	21.58 ± 0.13	34.06 ± 0.15	17.65 ± 0.10	26.55 ± 0.14

Table 2 presents the average prediction results of different algorithms for traffic flow forecasting over the next hour. As shown in Table 2, Seq2-ResGCRN achieves competitive performance compared with the other eight comparison methods (HA, VAR, SVR, LSTM,

GRU, STGCN, TCN, and ASTGCN) on both datasets in our experiments. Among them, the methods based on statistical approaches and traditional machine learning yield relatively lower performance in terms of evaluation metrics, indicating their limited capability in capturing the nonlinear and complex temporal patterns of traffic flow. Among the deep learning-based methods, LSTM, GRU, and TCN yield relatively poor prediction results, as they only consider the temporal correlations of traffic flow with relatively simple architectures. These models are prone to overfitting during training, leading to a significant deviation between learned traffic patterns and real-world dynamics. In contrast, models that integrate both spatial and temporal features achieve higher prediction accuracy than those relying solely on temporal features. Seq2-ResGCRN outperforms STGCN and ASTGCN, both of which also adopt spatio-temporal feature fusion. Notably, STGCN and ASTGCN still rely on the road graph structure based on geographic neighbor relationships for graph convolution operations, which limits their predictive performance to some extent.

To further demonstrate the advantages of the Seq2-ResGCRN model over the other comparison algorithms, Figure 6 and Figure 7 illustrate the MAE values at each prediction step for different algorithms on the PEMS08 and PEMS04 datasets, respectively. As can be observed, among the deep learning-based methods, LSTM, GRU, and TCN exhibit relatively poor overall performance, with unsatisfactory results across all prediction steps. VAR and SVR achieve competitive performance when the prediction horizon is less than 2, but their performance degrades significantly as the prediction horizon increases, indicating their limited generalization capability for long-term forecasting. Although Seq2-ResGCRN performs relatively less effectively for short-term predictions compared with some baselines, its predictive performance remains relatively stable, and the performance gap between Seq2-ResGCRN and the other methods tends to widen as the prediction horizon extends in our experiments.

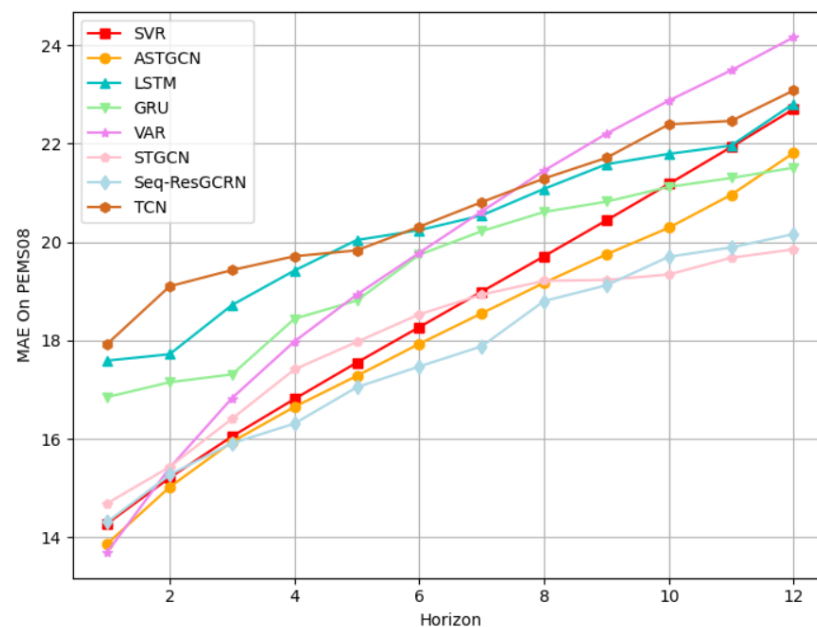


Figure 6. MAE results at different prediction steps on the PEMS08 dataset.

To further investigate the contribution of each module in the Seq2-ResGCRN model, we design two variants of Seq2-ResGCRN. Variant (a) excludes the adaptive graph learning module, i.e., it uses the road network graph A based on geographic neighbor relationships combined with Seq2-ResGCRN. Variant (b) replaces the ResGCRN architecture with GCN and GRU layers, denoted as $\tilde{S} + \text{Seq2-GCN-GRU}$. Variant (c) is the complete Seq2-

ResGCRN model, which integrates the adaptive graph learning module with the ResGCRN architecture. We evaluate these two variants together with the full Seq2-ResGCRN model on the PEMS04 and PEMS08 datasets. The experimental results are summarized in Table 3.

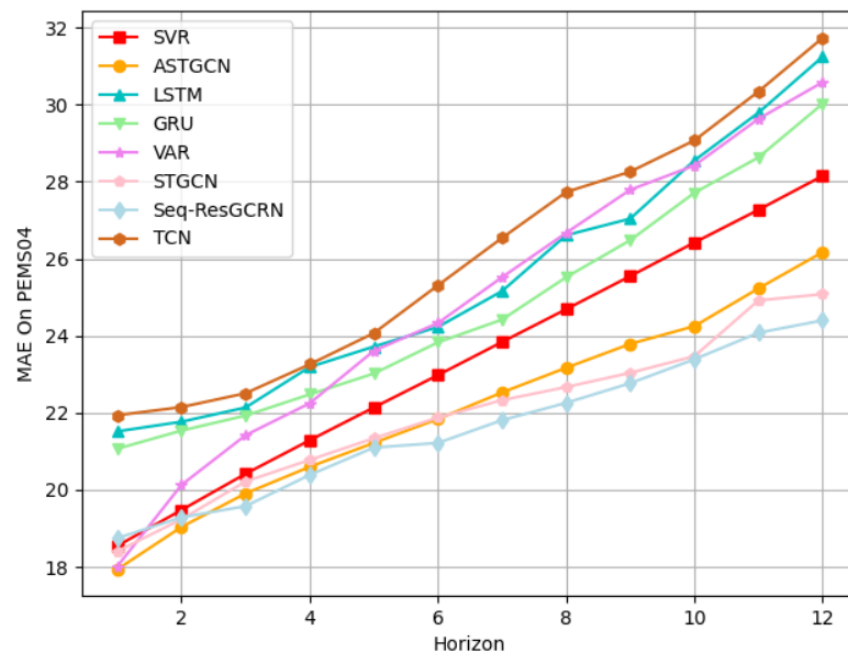


Figure 7. MAE results at different prediction steps on the PEMS04 dataset.

Table 3. Ablation study results of different modules in Seq2-ResGCRN.

Method	PEMS04		PEMS08	
	MAE	RMSE	MAE	RMSE
A+Seq2-ResGCRN (a)	22.04	35.06	17.92	26.98
$\tilde{S}$ +Seq2-GCN-GRU (b)	21.92	34.69	17.84	26.79
Seq2-ResGCRN (c)	21.58	34.06	17.65	26.55

As shown in Table 3, Seq2-ResGCRN achieves higher prediction accuracy than the other two variants. Comparing experiments (a) and (c), it can be observed that graph convolution with the sparse weighted graph generated by the adaptive graph learning module outperforms that with the adjacency matrix derived from geographic neighbor relationships. This provides evidence that the adaptive graph learning module can effectively optimize the spatial graph structure of the road network. Furthermore, comparing experiments (b) and (c), the ResGCRN module achieves superior performance over the stacked GCN and GRU layers, indicating that the ResGCRN architecture enhances the learning capability of GCN and GRU, and better captures the spatio-temporal dependencies of short-term traffic flow.

To highlight the effectiveness of Seq2-ResGCRN, we select a specific sensor node from each of the two datasets for visual analysis, and the results are shown in Figures 8 and 9. It can be observed that short-term traffic flow exhibits certain periodic patterns, with significant differences between peak and off-peak hours. On both the PEMS04 and PEMS08 datasets, the predicted curves of our model closely match the ground-truth short-term traffic variations, which confirms the effectiveness of the proposed model.

More specifically, we select one day from the test set to compare the predicted values with the ground-truth values, and the results are shown in Figures 10 and 11. It can be observed that our model still achieves satisfactory fitting performance on short-term

traffic flow sequences, which are nonlinear, non-stationary, and exhibit complex variations. The predicted curves generally capture actual traffic dynamics, demonstrating the effectiveness of our model.

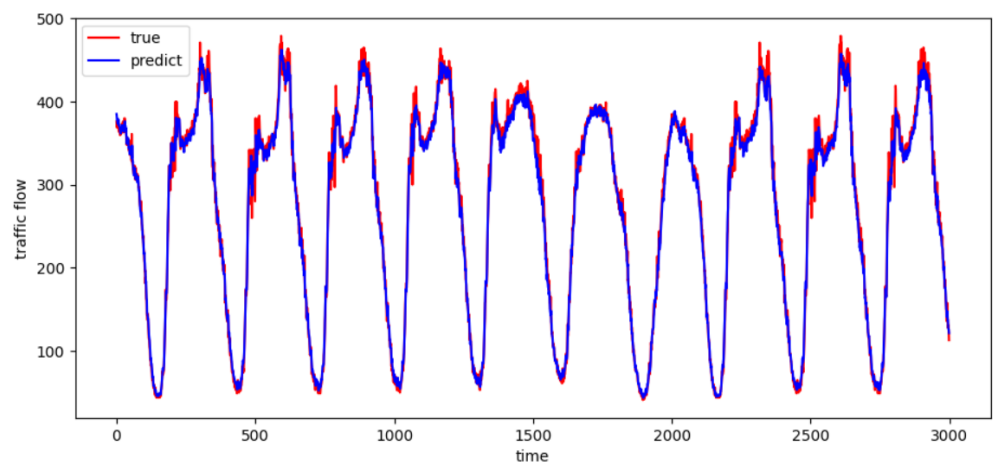


Figure 8. Visualization of traffic data on the PEMS08 dataset over a certain period.

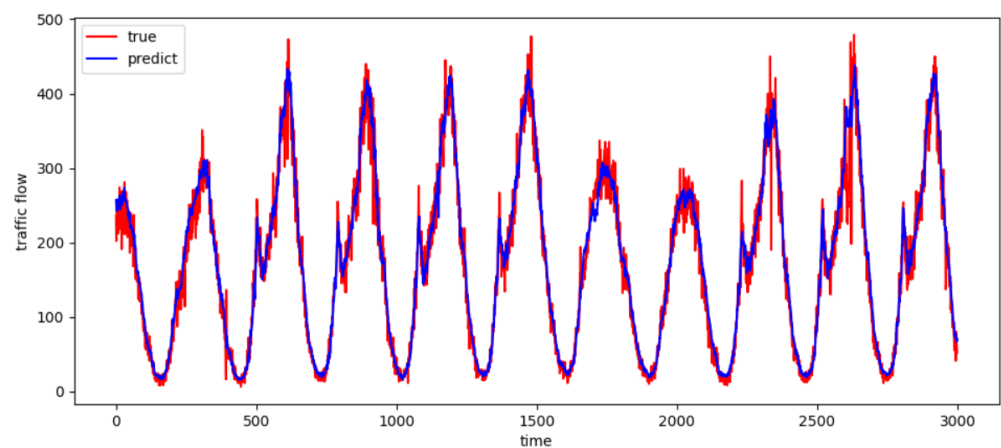


Figure 9. Visualization of traffic data on the PEMS04 dataset over a certain period.

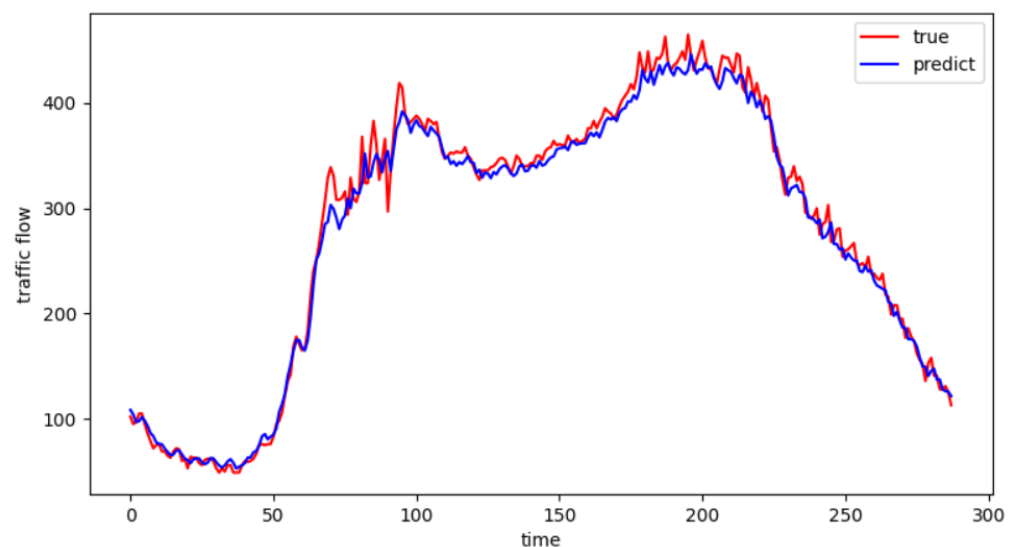


Figure 10. Visualization of traffic data on the PEMS08 dataset for one day.

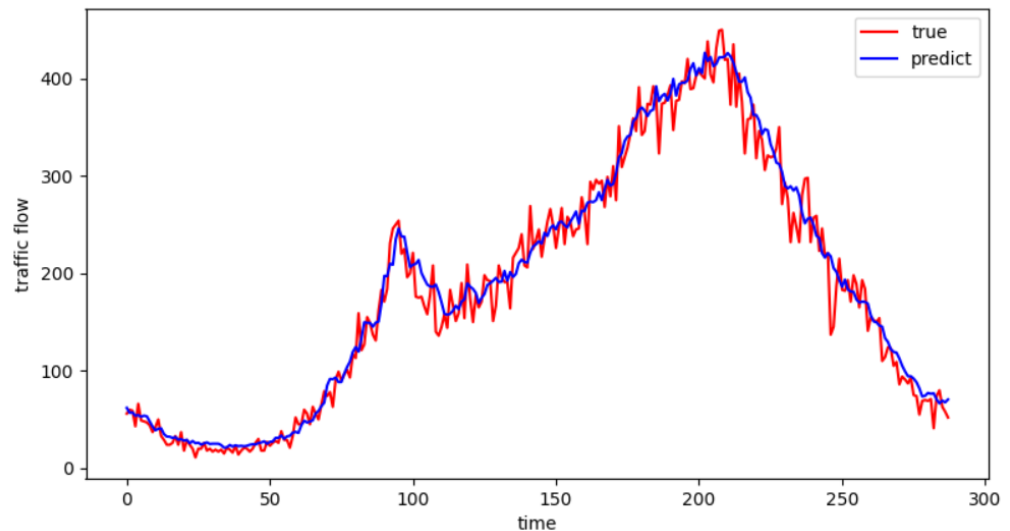


Figure 11. Visualization of traffic data on the PEMS04 dataset for one day.

6. Conclusions

In this paper, we propose a novel short-term road traffic flow prediction model, termed Seq2-ResGCRN, which employs an adaptive adjacency matrix combined with graph diffusion convolution to generate the road network graph. Compared with the graph structure derived from geographic locations, the graph generated by the adaptive approach better reflects the relative positional relationships among nodes in a real road network, enabling the extraction of richer spatial features during graph convolution. Furthermore, in contrast to the simple stacking of GCN and GRU layers, the proposed ResGCRN architecture more effectively captures the spatio-temporal correlations of short-term traffic flow. Experimental results on two large-scale public datasets demonstrate that Seq2-ResGCRN achieves reasonably satisfactory prediction accuracy against the compared baseline models, providing some supporting evidence for the effectiveness of the proposed method. It should be noted that the conclusions drawn in this study are limited to the specific models and datasets evaluated. Future work will further examine the applicability of the proposed method across a broader range of models and scenarios.

Author Contributions: Conceptualization, W.Y. and T.L.; investigation, W.Y. and T.L.; methodology, W.Y. and T.L.; resources, W.Y.; software, W.Y. and T.L.; validation, T.L.; writing—original draft, W.Y. and T.L.; writing—review and editing, W.Y. and T.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research Foundation of Education Bureau of Hunan Province, China (Grant No.25B0864) and the Natural Science Foundation of Hunan Province, China (Grant No.2026JJ81126).

Data Availability Statement: The data presented in this study are openly available in IEEE Dataport at <https://iee-dataport.org/documents/pems04-and-pems08-traffic-flow-datasets-traffic-flow-forecasting> (accessed on 30 July 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations and key parameters are used in this manuscript:

Abbreviations

LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit

ITS	Intelligent Transportation System
IoT	Internet of Things
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
TCN	Temporal Convolutional Network
GDC	Graph Diffusion Convolution
FC	Connected Layer
Key Parameters	
V	Set of nodes corresponding to traffic sensors
N	Total number of nodes in the network
E	Set of edges representing connectivity among sensors
A	Adjacency matrix that encodes the graph topology
X	Traffic flow matrix
T	Number of time steps
$x_{t-k:t-1}$	Historical traffic sequence of length k
$\hat{x}_{t:t+h-1}$	Predicted sequence for the future h time steps
$\hat{A}_{adp}$	Adaptive adjacency matrix
θ_k	Weighting coefficient
λ_i	Eigenvalues of T
S	Graph diffusion matrix
D	Degree matrix
GC	Graph convolution operation
X_t	Input at time t
$\tilde{S}$	Sparse graph structure
H	Hidden state information

References

1. Ceder, A. Urban mobility and public transport: Future perspectives and review. *Int. J. Urban Sci.* **2021**, *25*, 455–479. [\[CrossRef\]](#)
2. Faheem, H.B.; Shorbagy, A.M.E.; Gabr, M.E. Impact of traffic congestion on transportation system: Challenges and remediations—A review. *Mansoura Eng. J.* **2024**, *49*, 18. [\[CrossRef\]](#)
3. Zhang, X.; Li, J.; Zhou, J.; Zhang, S.; Wang, J.; Yuan, Y.; Liu, J.; Li, J. Vehicle-to-everything communication in intelligent connected vehicles: A survey and taxonomy. *Automot. Innov.* **2025**, *8*, 13–45. [\[CrossRef\]](#)
4. Cheng, Z.; Pang, M.S.; Pavlou, P.A. Mitigating traffic congestion: The role of intelligent transportation systems. *Inf. Syst. Res.* **2020**, *31*, 653–674. [\[CrossRef\]](#)
5. Yin, X.; Wu, G.; Wei, J.; Shen, Y.; Qi, H.; Yin, B. Deep learning on traffic prediction: Methods, analysis, and future directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 4927–4943. [\[CrossRef\]](#)
6. Wang, Z.; Sun, P.; Hu, Y.; Boukerche, A. A novel hybrid method for achieving accurate and timeliness vehicular traffic flow prediction in road networks. *Comput. Commun.* **2023**, *209*, 378–386. [\[CrossRef\]](#)
7. Sayed, S.A.; Abdel-Hamid, Y.; Hefny, H.A. Artificial intelligence-based traffic flow prediction: A comprehensive review. *J. Electr. Syst. Inf. Technol.* **2023**, *10*, 13. [\[CrossRef\]](#)
8. Sun, P.; Aljeri, N.; Boukerche, A. Machine learning-based models for real-time traffic flow prediction in vehicular networks. *IEEE Netw.* **2020**, *34*, 178–185. [\[CrossRef\]](#)
9. Xu, W.; Zhou, H.; Cheng, N.; Lyu, F.; Shi, W.; Chen, J.; Shen, X. Internet of vehicles in big data era. *IEEE/CAA J. Autom. Sin.* **2017**, *5*, 19–35. [\[CrossRef\]](#)
10. Aouedi, O.; Le, V.A.; Piamrat, K.; Ji, Y. Deep learning on network traffic prediction: Recent advances, analysis, and future directions. *ACM Comput. Surv.* **2025**, *57*, 1–37. [\[CrossRef\]](#)
11. Xu, W.; Liu, J.; Yan, J.; Yang, J.; Liu, H.; Zhou, T. Dynamic spatiotemporal graph wavelet network for traffic flow prediction. *IEEE Internet Things J.* **2023**, *11*, 8019–8029. [\[CrossRef\]](#)
12. Han, J.; Zhang, W.; Liu, H.; Tao, T.; Tan, N.; Xiong, H. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proc. VLDB Endow.* **2024**, *17*, 1081–1090. [\[CrossRef\]](#)
13. Ali, A.; Naeem, H.Y.; Sharafian, A.; Qiu, L.; Wu, Z.; Bai, X. Dynamic multi-graph spatio-temporal learning for citywide traffic flow prediction in transportation systems. *Chaos Solitons Fractals* **2025**, *199*, 116898. [\[CrossRef\]](#)
14. Ali, A.; Ullah, I.; Ahmad, S.; Wu, Z.; Li, J.; Bai, X. An attention-driven spatio-temporal deep hybrid neural networks for traffic flow prediction in transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 14154–14168. [\[CrossRef\]](#)

15. Lu, Y.; Huo, G.; Cui, X.; Wang, B.; Zhang, Y.; Cui, Z. Emergency Events Traffic Flow Forecasting Using Text-Prompt-Guided Multimodal Large Language Models. *IEEE Trans. Intell. Transp. Syst.* **2026**, *27*, 8526–8538. [\[CrossRef\]](#)
16. Yan, F.; Wang, H. State-DynAttn: A Hybrid State-Space and Dynamic Graph Attention Architecture for Robust Air Traffic Flow Prediction Under Weather Disruptions. *Mathematics* **2025**, *13*, 3346. [\[CrossRef\]](#)
17. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Li, K.; Zeng, Z. Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **2020**, *14*, 1–23. [\[CrossRef\]](#)
18. Sui, G.; Song, M.; Sha, J.; Sun, H.; Yang, Q.; Xu, Z. Graph-Based Deep Learning for Short-Term Traffic Flow Forecasting on Urban Arterial Segments. In Proceedings of the 2025 8th International Conference on Computer Information Science and Artificial Intelligence, Wuhan, China, 12–14 September 2025; pp. 1569–1573.
19. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
20. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17804–17815.
21. Shin, Y.; Yoon, Y. PGCN: Progressive graph convolutional networks for spatial-temporal traffic forecasting. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 7633–7644. [\[CrossRef\]](#)
22. Li, M.; Zhu, Z. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February; Volume 35, pp. 4189–4196.
23. Rao, X.; Wang, H.; Zhang, L.; Li, J.; Shang, S.; Han, P. FOGS: First-Order Gradient Supervision with Learning-based Graph for Traffic Flow Forecasting. In Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022; pp. 3926–3932.
24. Guo, K.; Hu, Y.; Sun, Y.; Qian, S.; Gao, J.; Yin, B. Hierarchical graph convolution network for traffic forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 151–159.
25. Shu, W.; Cai, K.; Xiong, N.N. A short-term traffic flow prediction model based on an improved gate recurrent unit neural network. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 16654–16665. [\[CrossRef\]](#)
26. Chauhan, N.S.; Kumar, N.; Eskandarian, A. A novel confined attention mechanism driven bi-gru model for traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 9181–9191. [\[CrossRef\]](#)
27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
29. Gasteiger, J.; Weißerberger, S.; Günnemann, S. Diffusion improves graph learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 13354–13366.
30. Page, L.; Brin, S.; Motwani, R.; Winograd, T. The pagerank citation ranking: Bring order to the web. In Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 14–18 April 1998.
31. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
32. Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; Jia, Z. Freeway performance measurement system: Mining loop detector data. *Transp. Res. Rec.* **2001**, *1748*, 96–102. [\[CrossRef\]](#)
33. Zivot, E.; Wang, J. Vector autoregressive models for multivariate time series. In *Modeling Financial Time Series with S-PLUS®*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 369–413.
34. Ho, C.H.; Lin, C.J. Large-scale linear support vector regression. *J. Mach. Learn. Res.* **2012**, *13*, 3323–3348.
35. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929.
36. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
37. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.