Scientific Research Publishing

Journal of Computer and Communications, 2025, 13(8), 163-180
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

# Traffic Speed Prediction Based on Autoencoder and Deep Learning

**Zhuowei Fu, Huifang Feng***

College of Mathematics and Statistics, Northwest Normal University, Lanzhou, China
Email: *hffeng@nwnu.edu.cn

**How to cite this paper:** Fu, Z.W. and Feng, H.F. (2025) Traffic Speed Prediction Based on Autoencoder and Deep Learning. *Journal of Computer and Communications*, **13**, 163-180.
https://doi.org/10.4236/jcc.2025.138008

**Received:** July 9, 2025
**Accepted:** August 12, 2025
**Published:** August 15, 2025



## Abstract

Traffic prediction is the core of intelligent transportation system, and accurate traffic speed prediction is the key to optimize traffic management. Currently, the traffic speed prediction model based on deep learning has become a research hotspot in the field of transportation. With the rapid development of deep learning and the improvement of computer hardware performance, traffic speed prediction based on deep learning has become a hot spot and mainstream of research. In this paper, a traffic speed prediction model based on autoencoder structure is proposed by combining Causal Convolutional Network (CCN), Graph Convolutional Network (GCN) and Multi-Head Self-Attention (MHSA). The model realizes efficient extraction and fusion of spatio-temporal features through a layered design: GCN handles spatial features, CCN and MHSA handle temporal features. First, in the encoder, multiple 2D causal convolution modules are utilized to capture the core features of traffic flow and remove redundant information. Second, the attention weights are dynamically computed using MHSA to identify important time points and sub-sequences in the traffic sequence, and the spatial features of the traffic flow captured using GCN. Further, when reconstructing potential features in the decoder, jump connections from the encoder are added, so that the decoder multiplexes the shallow features extracted by the encoder in the feature reconstruction stage and retains more detailed information of the original data. Finally, the prediction results are obtained by nonlinear fusion of the autoencoder information by the fully connected network. The experimental results show that compared with many baseline models, the proposed model in this paper is able to capture the spatio-temporal correlation of traffic speed data in traffic flow prediction and has good prediction performance.

## Keywords

Traffic Speed Prediction, Causal Convolutional Network, Multi-Head Self-Attention, Graph Neural Network, Autoencoder

## 1. Introduction

In the face of such a severe urban traffic predicament, the significance of traffic flow prediction as the cornerstone of urban traffic control and guidance has become more and more prominent. Traffic flow prediction through the future period of time on the road vehicle flow, speed, density and other parameters for the scientific prediction, for traffic management departments to provide a basis for decision-making. For example, in advance to grasp the changes in a road traffic flow, you can reasonably plan the traffic signal timing, to avoid excessive gathering of vehicles; for the possible congestion of the road, timely release of traffic warning information, to guide the public to choose the optimal travel route. Accurate traffic flow prediction is the core of optimized traffic management, which can help traffic management departments to achieve efficient allocation of traffic resources, enhance road capacity and ease traffic congestion.

In recent years, with the rapid development of artificial intelligence technology, the traffic flow prediction model based on deep learning has been generated, and has rapidly become a research hotspot in the field of transportation. Deep learning, with its powerful feature extraction and data processing capabilities, can automatically mine the complex spatial and temporal features in traffic data, effectively deal with the complexity of traffic flow, and provide a more accurate and reliable solution for traffic flow prediction, thus opening up new paths for easing urban traffic congestion and optimizing traffic management.

Currently, traffic flow prediction methods are broadly categorized into three types, *i.e.*, traditional time series methods, machine learning methods and deep learning methods. Traditional time series methods mainly focus on the time-series information contained in the data, using methods such as Kalman filter analysis (KF) [1], History Average (HA) [2], Vector Autoregression (VAR) [3], Autoregression Integrated Moving Average Model (ARIMA) [4], etc., which predict future traffic conditions through simple mathematical operations. The advantage of traditional time series methods is that the model is simple and the calculation speed is fast; the disadvantage is that only smooth time series data can be considered, which is not applicable to nonlinear traffic flow data, and only the temporal information in the traffic flow can be extracted, and the spatial dependence contained therein cannot be modeled.

In order to solve the limitations of traditional statistical models, many machine learning models have been widely used by researchers in the field of traffic state prediction, such as K-Nearest Neighbor (KNN) [5], Support Vector Machine (SVM) [6], Random Forest (RF) [7], Feedforward Neural Network (FNN) [8] and other machine learning methods. Literature [9] discusses in detail the main factors affecting traffic, experiments on traditional machine learning based traffic flow prediction models on multiple datasets, and comparatively analyzes the prediction performance of each model. The experimental results show that these data-driven machine 2 W. S. Khedr learning-based methods can better capture the nonlinear dependencies in the data and achieve better prediction accuracy in short-term

traffic flow prediction. However, traditional machine learning methods are difficult to effectively extract features from high-dimensional complex traffic data, and cannot tap into the deep and implicit spatio-temporal correlations in big data, and thus are difficult to be scaled up and applied to large-scale real-world traffic prediction applications [10].

In recent years, the rapid development of deep learning technology has injected new vigor into various industries. Deep learning models have gradually become the mainstream method in the field of traffic state prediction due to their excellent performance in capturing the complex structural attributes of data. Currently, the deep learning network models commonly used for traffic flow prediction are Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Graph Convolutional Network (GCN), and Transformer [11]-[13], etc. Jiang *et al.* [14] proposed spatio-temporal meta-graph learning, a novel graph structure learning method, which was applied to a meta-graph convolutional recurrent network to achieve robust prediction performance. Yin *et al.* [15] proposed the bi-directional gated recurrent unit (BiGRU), which was combined with the Improved Variational Modular Decomposition (VMD), Graph Attention Network (GAT) to from a spatio-temporal hybrid prediction model ST-VGBiGRU, which is able to train the optimal parameters of the model using the improved RMSprop algorithm while improving the prediction accuracy of the model. Yu *et al.* [16] further combined temporal convolution with spatial convolution and proposed the STGCN model, which performs excellently in nonlinear feature modeling. Ji *et al.* [17] proposed the Spatio-Temporal Self-Supervised Learning (STSSL), which is able to achieve adaptive heterogeneity perception of noise perturbations on spatio-temporal graphs. Qi *et al.* [18] proposed the Spatial–Temporal Fusion Graph Neural Network, STFGCN, which constructs dynamic adaptive graphs for modeling by extracting multi-scale temporal dependencies from multiple semantic environments. Kong *et al.* [19] proposed the Spatio-Temporal Pivotal Graph Neural Network (STPGNN), which is capable of identifying key nodes and accurately capturing spatio-temporal dependencies centered on key nodes, in addition to extracting spatio-temporal traffic features on key and non-key nodes through a parallel framework.

After the rise of the Transformer architecture, Jiang *et al.* [20] proposed the PDFormer model, which combines the propagation delay property with the dynamic attention mechanism to effectively solve the problem of modeling long time series. Ma *et al.* [21] proposed a framework that relies entirely on the original Transformer architecture, and designed the embedding module to extract the spatio-temporal dependencies of traffic flow, in addition to using a pre-trained language model to improve the prediction performance. Shi *et al.* [22] proposed a lightweight traffic flow prediction model, TFPformer, based on the Transformer model, which employs a parallel structure to capture spatio-temporal dependencies in traffic flow. Moon *et al.* [23] proposed a subgraph-based graphical Transformer traffic flow prediction model, which is capable of capturing spatial heter-

ogeneity and can effectively handle the long-term temporal dependence of traffic flow, and the model validity was verified in four traffic benchmark tests. Li *et al.* [24] modeled traffic flow as a diffusion process on a directed graph, 3W. S. Khedr introduced a diffusion convolutional recurrent neural network, DCRNN, to capture spatial dependence through bi-directional stochastic wandering on the graph and used an encoder-decoder architecture to capture temporal dependence, and evaluated the results to show an improvement over the benchmark model. Zheng *et al.* [25] proposed Graph Multi-Attention Network GMAN, which reduces the error in traffic prediction by establishing a link between historical and future time steps through an attentional approach.

With the rapid development of deep learning and the improvement of computer hardware performance, traffic flow prediction based on deep learning has become a research mainstream. However, existing models still face critical gaps: many struggle to simultaneously filter redundant spatio-temporal features in high-dimensional traffic data, leading to noise interference; others fail to capture directional temporal dependencies (e.g., rush-hour propagation patterns) or dynamically prioritize crucial time steps and road segments, resulting in suboptimal fusion of spatio-temporal correlations. In this paper, a traffic flow prediction model based on autoencoder structure is proposed by combining CCN, GCN and MHSA. The model realizes efficient extraction and fusion of spatio-temporal features through layered design, with its specific combination addressing the aforementioned gaps: the autoencoder compresses redundant information and retains core features via encoding-decoding, mitigating noise issues; CCN captures directional temporal dependencies that traditional models overlook; GCN models the road network topology and extracts the spatial features of the data; and MHSA dynamically weights critical spatio-temporal points, enhancing the adaptability of feature fusion. This synergistic integration enables the model to capture the spatio-temporal correlation of traffic speed data more comprehensively and achieve good prediction performance.

## 2. Description of the Problem

### 2.1. Transportation Network Construction

Let $G_t = (V, E, A, X_t)$, $t = 1, 2.\cdots T$, where $G_t$ is a dynamic complex network, $V$ and $E$ are sets of nodes and edges respectively, $|V| = N$ is the number of nodes in the graph, $A = (a_{ij})_{N\times N}$ is the adjacency matrix. If $v_i$, $v_j \in V$, $(v_i, v_j) \in E$, $a_{ij} = 1$. Others $a_{ij} = 0$. $X_t \in R$ is the matrix of characteristic attributes. It represents the traffic flow information of the traffic network in time period t, such as traffic volume, traffic speed, and traffic density. When constructing a traffic network, the nodes of the network can be determined according to the data flow acquisition method, for example, if the traffic flow data is obtained through multiple sensors distributed on the road network, the sensors are viewed as network nodes to construct the traffic network. If the traffic flow data is obtained through vehicle GPS, the intersection can be viewed as a network node to construct the traffic

network, *i.e.*, the network can be constructed by using the master method, or by using the pairwise method to construct the traffic network [26].

### 2.2. Definition of the Problem

Historical observations of the transportation network are known for the past $p$ time periods $\left\{G_{t-p+1},\ G_{t-p+2},\cdots,G_t\right\}$ are known to predict the transportation network data for the future $q$ time periods $\left\{G_{t+1},\ G_{t+2},\cdots,G_{t+q}\right\}$. The traffic flow prediction problem can be defined as:

$$\left[G_{t+1},\ G_{t+2},\cdots,G_{t+q}\right]=f\left(G_{t-p+1},\ G_{t-p+2},\cdots,G_t\right) \tag{1}$$

where $f(g)$ is a learnable prediction function. When $q=1$ is a single-step prediction and $q\geq 2$ is a multi-step prediction.

## 3. Traffic Speed Prediction Based on Autoencoder and Deep Learning

### 3.1. General Framework of the Model

In this paper, we propose a traffic prediction model based on autoencoder structure, whose core modules include Causal Convolutional Network (CCN), Multi-Head Self-Attention (MHSA) and Graph Convolutional Network (GCN), the overall framework of the model is shown in **Figure 1**, and the specific steps of traffic flow prediction include:

Step 1: **Input** The traffic network is constructed by preprocessing the attribute features of the traffic network nodes (*i.e.*, traffic speed).

Step 2: **Encoder** It consists of two causal convolution-pooling layers sequentially connected, each consisting of a causal convolution, an activation function SiLU, and a Dropout layer, which achieves dimensionality compression by pooling.

Step 3: **MHSA** For the features extracted by the encoder, the attention weights are adaptively assigned, the attention coefficients are computed, the importance of different time steps is learned, and the information of the time step that is most important for the current task is highlighted to capture long-term dependencies.

Step 4: **GCN** The output of the multi-attention mechanism is used as the input to the graph convolution, and the graph structure information is utilized to further extract features, and the feature representations of the nodes are updated by computing the information transfer between the nodes.

Step 5: **Decoder** Sequential connection of two causal convolution-pooling layers, recovering dimensionality by Up-pooling and using jump connections from encoder, allows the decoder to reuse shallow features extracted by the encoder in the feature reconstruction phase, preserving more detailed information of the original data. Specifically, the jump connections are implemented by addition: the second layer of the encoder is added to the first layer of the decoder, and the resulting output is fed into the second layer of the decoder; meanwhile, the first layer of the encoder is added to the second layer of the decoder. The output is passed

through the fully connected layer to obtain the final prediction.

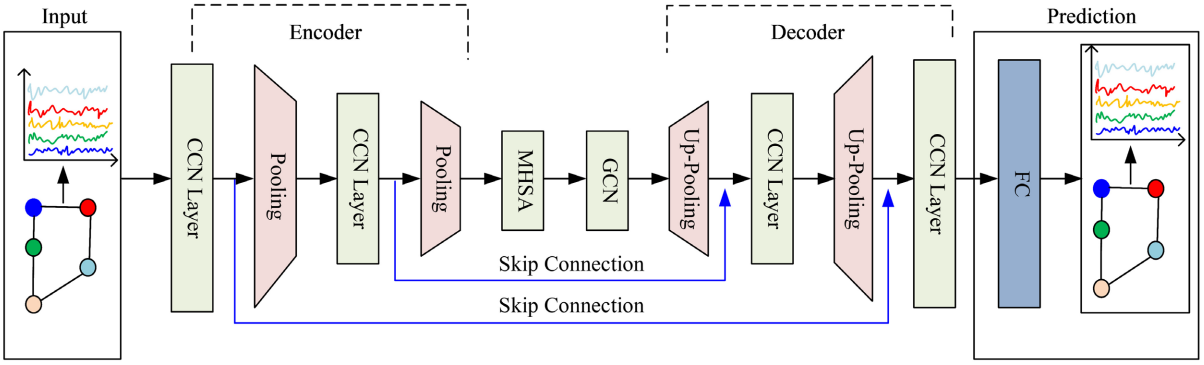



Figure 1. General framework of the model.

## 3.2. Autoencoder

Autoencoder is a neural network model that learns an efficient representation of data through an unsupervised learning approach [27]. It usually consists of an encoder and a decoder. The encoder usually consists of a series of neural network layers that achieve data compression by gradually reducing the number of neurons, *i.e.*, mapping the input data to a low-dimensional potential space. Decoder is constructed in contrast to an encoder and reconstructs the output similar to the input data by gradually increasing the number of neurons, *i.e.*, mapping the potential space representation back to the original data space. The autoencoder construction is shown in Figure 2.

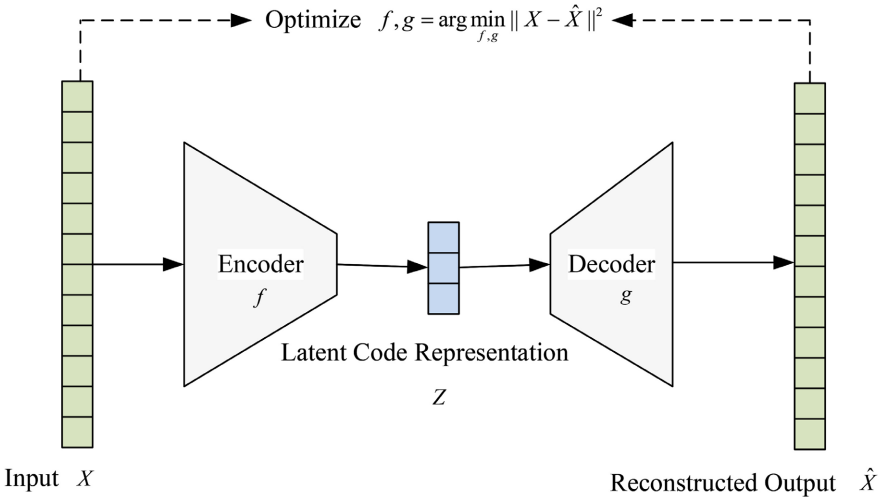



Figure 2. Autoencoder.

Let $X$ be the input data. $Z$ is the coded representation of the input data (potential space). $\hat{X}$ is the reconstructed output (decoded data). $W_e$ and $W_d$ are the encoder and decoder weight matrices. $b_e$ and $b_d$ are bias vectors. The encoder and decoder functions are mathematically defined as:

$$Z = f(X) = \sigma(W_e X + b_e) \tag{2}$$

$$\hat{X} = g(Z) = \sigma(W_d Z + b_d) \tag{3}$$

where $\sigma(\ )$ is activation function.

Autoencoder solves two functions $f$ and $g$. So that the input data and reconstructed output errors are minimized:

$$f, g = \arg \min_{f,g} \left\| X - \hat{X} \right\|^2 \tag{4}$$

### 3.3. Causal Convolution Network

Causal Convolution Network [28] is a special convolution operation in convolutional neural network that follows a causal convolution operation in the time or sequence dimension. When processing sequence data, it ensures that when calculating the output of a certain moment, only the input information of that moment and its predecessors will be used, and not the input information of future moments, which avoids the problem of information leakage, and conforms to the causal logic of processing sequence data in practical applications. A schematic diagram of one-dimensional CCN computation is shown in Figure 3.

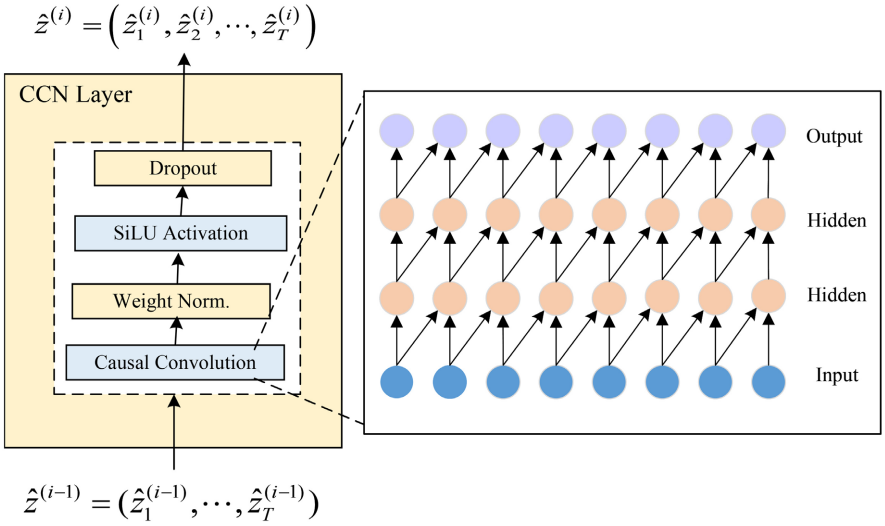



Figure 3. Causal convolution network layer.

Given an input one-dimensional time series $X = [x_1, x_2, x_3, \cdots, x_n]$ and $F = [f_1, f_2, f_3, \cdots, f_k]$ is the convolution kernel. The causal convolution operation $F$ is defined as:

$$(F * X)_t = \sum_{k=1}^{K-1} f_k x_t - K + k \tag{5}$$

where * denotes the dilation convolution operator.

Transportation time series data has both time dimension and spatial dimension information, so it is processed with two-dimensional causal convolution. Let the

input feature map be $X = \left(X_{i,j}\right)_{T\times S}$, where $T$ is the length of the time dimension and $S$ is the length of the space dimension. The convolution kernel is $K = \left(K_{i,j}\right)_{K_t\times K_s}$, where $K_t$ is the convolutional kernel time dimension and $K_s$ is the convolutional kernel space dimension. The bias is $b \in R$. To ensure causality, zero-padding is performed from the beginning of the time dimension. Suppose that the time dimension is populated with $p_t = K_{t-1}$ zeros. The filled feature map is $\tilde{X} = \left(\tilde{x}_{i,j}\right)_{(T+K_{t-1})}^{S}$. The two-dimensional causal convolution operation is:

$$\left(K * X\right)_{(t,s)} = \sum_{i=1}^{K_t}\sum_{j=1}^{K_s} k_{i,j}\tilde{x}_{t+i,s+j} + b \tag{6}$$

where * is the convolution operation. When $t+1 > T + K_{t-1}$ or $s + j > S$, $\tilde{x}_{t+i,s+j} = 0$.

The convolution operation is followed by a nonlinear transformation through activation functions, and the commonly used activation functions include Sigmoid, ReLU, SiLU, Tanh, and so on. In order to avoid overfitting the model on the training data, which leads to poor performance on the test data, the data is processed by the Dropout layer after the activation function. A portion of neurons and their connections are randomly dropped during the training process so that the model cannot rely on a specific combination of neurons, thus reducing the co-adaptation between neurons, forcing the network to learn more robust feature representations, improving the model's generalization ability, and reducing the risk of overfitting.

### 3.4. Multi-Head Self-Attention

The traffic flow of a node is not only affected by the historical traffic flow of that node, but also by the traffic flow of its neighboring nodes. Therefore, a spatio-temporal attention mechanism can be used to capture the dynamic spatio-temporal correlation of traffic flow, where the spatial attention mechanism capture the dynamic spatial correlation between a node and its neighboring nodes, and the temporal attention mechanism capture the dynamic temporal correlation between nodes at different times.

The self-attention mechanism portrays the interdependence between sequence elements by calculating the correlation (*i.e.*, weight) of each element in the sequence with all other elements. Multi-Head Self-Attention (MHSA) [29] capture the distribution of attention in time series data in different subspaces from multiple perspectives by introducing multiple heads of attention, thus enabling a better understanding of the interactions between variables in complex series data, and improving the model representation and generalization capabilities.

The steps to achieve MHSA include:

Step 1: **Input transformations** The input sequence $X$ is passed through three different linear transformations to obtain the query vector $Q$, the key vector $K$ and

the value vector $V$: $Q = XW_Q$, $K = XW_K$, $V = XW_V$, where $W_Q$, $W_K$, $W_V$ are the learnable weight matrices of $Q$, $K$, $V$.

Step 2: **Parallel computation of multiple attentions** For each head, the attention score was computed separately using dot product attention:

$$\text{Head}_i = \text{Attention}\left(Q_i, K_i, V_i\right) = \text{SoftMax}\frac{Q_i K_i^T}{\sqrt{d_k}} V_i \tag{7}$$

where $d_k$ is the embedding dimension.

Step 3: **Splicing the attention heads and linearly transforming them output** The spliced vectors are passed through a linear transformation to get the final multi-head attention data:

$$\text{MultiHead}\left(Q, K, V\right) = \text{Concat}\left(\text{head}_1, \cdots, \text{head}_h\right) W_o \tag{8}$$

where $W_o$ is the output weight matrix and $h$ is the number of attention heads. The structure of MHSA is shown in Figure 4.

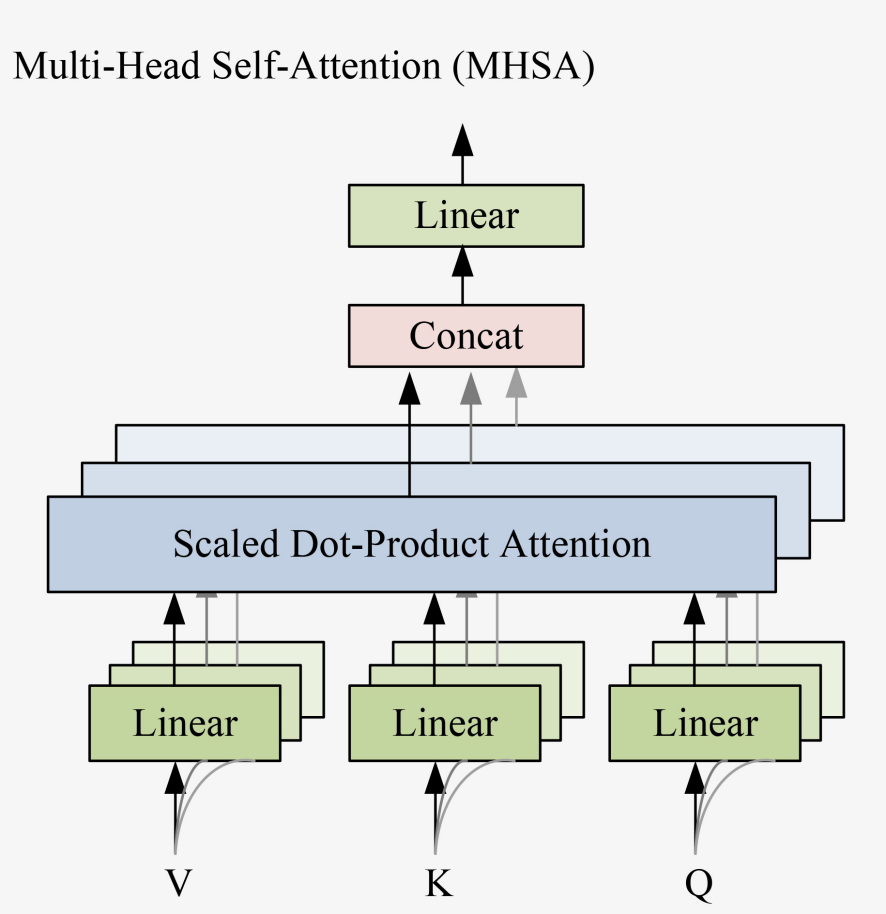



Figure 4. Multi-Head self-attention.

### 3.5. Graph Convolutional Network

Graph convolutional network [30] is a neural network structure for deep learning. Compared with the traditional network model CNN for deep learning, graph con-

volutional network is able to handle spatio-temporal data with non-Euclidean structure. Traffic road network is a typical complex network structure, and its network characteristics are inscribed with non-Euclidean structure data, using graph convolutional network can extract the topological relationship between nodes and their surrounding nodes in the complex network, so as to accurately extract the spatial characteristics of the traffic data, and lay the foundation for improving the accuracy of traffic speed prediction.

Graph convolution usually includes null-domain convolution and spectral-domain convolution, the core of null-domain convolution is the aggregation process of neighboring node feature and inference to get the prediction, such as GraphSage network (2017). Spectral domain convolution is network learning by neighbor matrix and node attribute features, such as Graph Convolutional Network (GCN). In this paper, we use GCN to construct a deep learning network. In the spectral domain map convolution, we convolve the graph $G$ with the convolution kernel $g_\theta$:

$$g_\theta *_G X = g_\theta(L)X = g_\theta\left(U\Lambda U^T\right)X = Ug_\theta(\Lambda)U^T X \tag{9}$$

where $*G$ is the graph convolution operation. Laplace $L = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}} = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, $D \in R^{N\times N}$ is the degree matrix, $I_N \in R^{N\times N}$ is the unit matrix and $X$ is the attribute feature matrix of the graph. The eigenvalue decomposition of $L$ is $L = U\Lambda U^T$, where $\Lambda$ is the diagonal matrix consisting of the eigenvalues of $L$ and $U = \{u_1, u_2, \cdots u_N\}$ consists of the eigenvectors of $L$.

Due to the large structure of the traffic road network graph, the computation of the Laplace transform on the graph usually involves complex operations such as eigen-decomposition of the Laplace matrix of the graph, which is computationally expensive. To avoid higher computational complexity, Chebyshev polynomials are used to obtain an efficient approximation, so the graph convolution can be rewritten as:

$$g_\theta *_G X \approx \sum_{k=0}^{K-1}\theta_k T_k\left(\tilde{L}\right)X \tag{10}$$

where $\theta \in R^K$ is a vector of polynomial coefficients and $K$ is the size of the graph convolution kernel. $T_k\left(\tilde{L}\right) \in R^{N\times N}$ is a term of order $k$ of the Chebyshev polynomial, which satisfies the recurrence formula: $T_k(X) = 2XT_{k-1}(X) - T_{k-2}(X)$, $T_0(X) = 1$, $T_1(X) = X$, $\tilde{L} = \frac{2L}{\lambda_{\max}} - I_N$, $\lambda_{\max}$ is the largest eigenvalue of $L$.

### 3.6. Loss Function

In this paper, Huberloss is chosen as the loss function for training, and the use of MSE for small errors and MAE for large errors is able to overcome the drawbacks of both by smoothing out the training process as well as minimizing the effect of

outliers. The mathematical formula of Huberloss is defined as follows:

$$L_\delta(y,\hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2, & |y-\hat{y}| \le \delta \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2, & \text{others} \end{cases} \tag{11}$$

where $y$ is the true value, $\hat{y}$ is the predicted value, $\delta$ is the threshold value. (In this paper, $\delta = 1$.)

## 4. Experimental Results and Analysis

### 4.1. Datasets Introduction and Preprocessing

In this paper, the validity of the model is tested using the PEMSD7M and Traffic SH datasets. PEMSD7M (http://pems.dot.ca.gov) were collected and provided by the Performance Measurement System (PeMS) of the California Department of Transportation. The dataset was selected from 228 sensor nodes from the transportation roadway network, and monitoring data was selected as a sample for the period from May 1, 2017 to August 31, 2017, with data collected at a frequency of every 5 minutes. TrafficSH selected 896 sensor nodes in Shanghai from March 5, 2022 to April 5, 2022, and the frequency of data collection was every 30 minutes. The data collected by each sensor each time includes features in three dimensions: flow rate, average speed and average occupancy. For the missing values present in the datasets, this paper employs missing value filling using periodic features of traffic flow. Normalization is done using Z-Score.

### 4.2. Experimental Setup

70% of the datasets was selected as the training set, 15% as the test set, and 15% as the validation set. Multi-step prediction is performed using historical 12-step traffic speed data, including future 3, 6, 12-step prediction. The model use AdamW optimizer with a learning rate of 0.001. The batch-size is 32 and the training period is 500 with an early stop mechanism. All experiments were performed on a machine equipped with NVIDIA GeForce 3080 Ti GPU and 128GB of RAM. Models were implemented using python 3.10.0.

### 4.3. Metric

Commonly used assessment metrics for traffic flow forecasting include Mean Absolute Error(MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error(MAPE). The definitions of the three assessment metrics are presented as follows:

1) Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{12}$$

where $y_i$ is the true value, $\hat{y}_i$ is the predicted value and $n$ is the number of the records. The following formulas have the same meaning. The range of the MAE

error is in the interval [0, +∞). The MAE value is taken to be 0 when the predicted and true values match exactly, and furthermore, the large the error, the large the value.

2) Root Mean Squared Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{13}$$

The range of the RMAE error is in the interval [0, +∞). Similar to the MAE, the RMSE value is taken to be 0 when the predicted and true values match exactly, and the large the error, the large the value.

3) Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{14}$$

The range of the MAPE error is in the interval [0, +∞). MAPE is presented as a percentage, with a value of 0% indicating a perfect model and a value greater than 100% indicating a poor model.

## 4.4. Results

### 4.4.1. Baseline

In this paper, we use History Average Model (HA), Feedforward Neural Network (FNN), Spatiotemporal Graph Convolutional Network (STGCN), Diffusion Convolutional Recurrent Neural Network (DCRNN), and Graph Multi-Attention Network (GMAN) for the comparison of the prediction performance with the proposed model (Ours).

HA [2]: A simple and basic time series prediction model, usually used to forecast data with cyclical or seasonal patterns. Its core assumption is that the data has a cyclical pattern of change in time. Based on this assumption, the model predicts future values by calculating the average value over the same time period in history.

FNN [8]: It is characterized by a unidirectional transfer of information from the input layer to the output layer, with a number of hidden layer in between, and the neurons in each layer are nonlinearly transformed by weighted sums and activation functions.

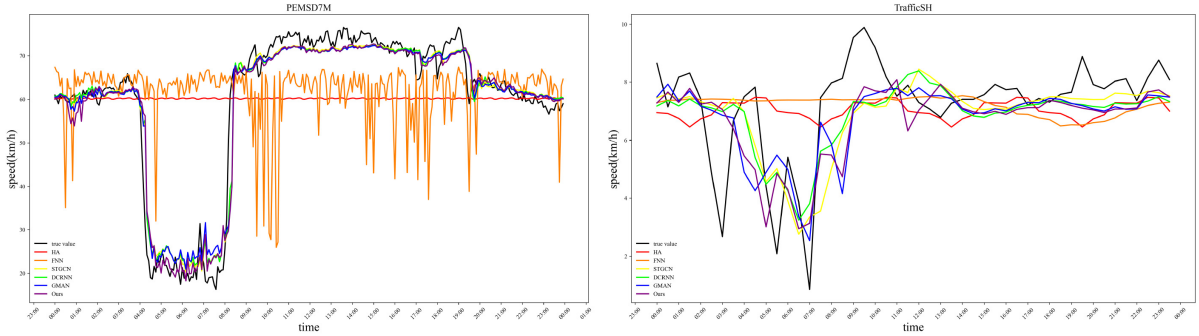
STGCN [16]: A deep learning model combining Graph Convolutional Network (GCN) and Convolutional Neural Network (CNN) is specialized to deal with spatio-temporal relationships in graph-structure data. It captures spatial dependencies by mapping time-series data onto graphs using graph convolution, while combining it with temporal convolution to handle temporal dependencies.

DCRNN [24]: It models the dynamics of traffic flow as a directed weighted graph and proposes diffusion convolution operations to capture spatial dependencies. The upstream and downstream traffic impacts are captured flexibly by incorporating bi-directional random wandering in the convolution.

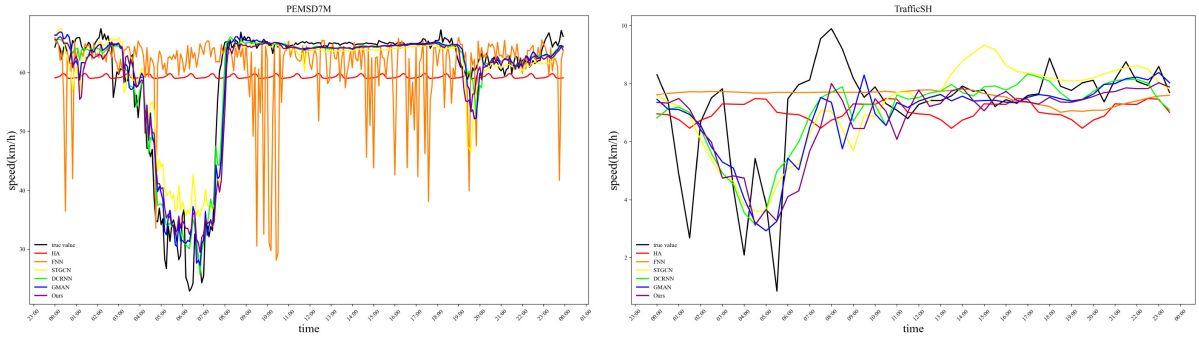
GMAN [25]: The model uses an encoder-decoder structure, where the encoder and decoder consist of multiple spatio-temporal attention blocks, each consisting

of a spatial attention mechanism, a temporal attention mechanism, and a gated fusion mechanism.

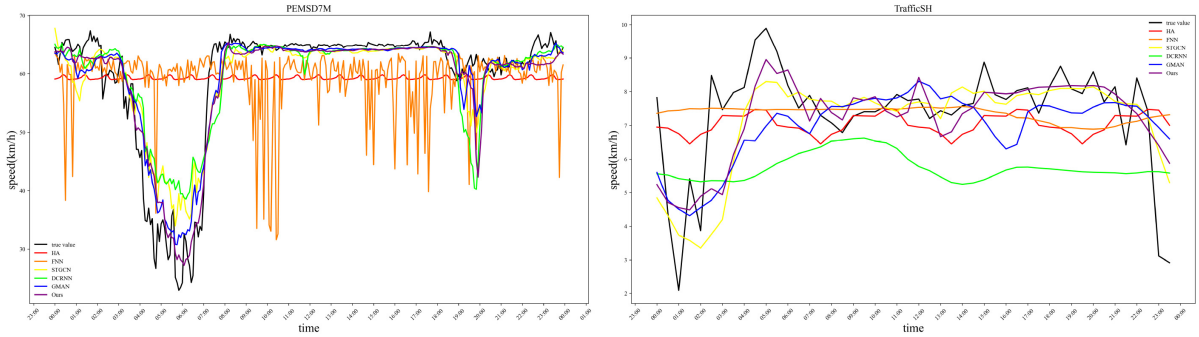
#### 4.4.2. Comparative Experiments and Analysis of Results




(a)3-step




(b)6-step




(c)12-step

**Figure 5.** Visualization of true values and predicted values.

Figure 5 shows the comparison of traffic speed prediction between the five baseline models and the model proposed in this paper (Ours), with the 24 hours prediction results from a randomly selected sensor visualized. Table 1 shows the prediction performance metrics of each model, where black bold indicates the

metrics with the best results, and underlining marks the metrics with the second highest results.

From Table 1, it can be seen that the HA and FNN models cannot well explore the mutual influence of traffic speeds between different road sections and at different time points, and lack of consideration of the spatial structure of the complex road network and the temporal correlation of traffic flow, and the prediction performances of these two models on both datasets are poorer. The STGCN, DCRNN, and GMAN models extract the features of the data in both the temporal and spatial dimension, and achieve better results on the two datasets. Both datasets with better results. Compared with the baseline model, the performance of the proposed model in this paper performs optimally. On the dataset TrafficSH, the values of three error metrics such as MAE, RMSE and MAPE are minimized for the multi-step prediction of the proposed model in this paper. On the dataset PEMSD7M, the value of other metrics are also minimized, except for the MAE of 6-step and 12-step, which are ranked second. The results fully demonstrate that the prediction model in this paper is able to capture the spatio-temporal characteristics of traffic flow simultaneously and has excellent prediction performance for traffic speed.

Table 1. Comparison of multi-step prediction performance of various models.

| Dataset | Step | Metric | HA | FNN | STGCN | DCRNN | GMAN | Ours |
|---|---|---|---|---|---|---|---|---|
| TrafficSH | 3 | MAE | 1.03 | 1.27 | 0.54 | 0.53 | 0.60 | **0.48** |
| | | RMSE | 1.37 | 1.69 | 0.91 | 0.88 | 0.97 | **0.78** |
| | | MAPE | 16.83 | 21.23 | 7.67 | 8.00 | 8.50 | **6.31** |
| | 6 | MAE | 1.03 | 1.37 | 0.62 | 0.61 | 0.68 | **0.60** |
| | | RMSE | 1.37 | 1.83 | 0.93 | 0.93 | 1.01 | **0.90** |
| | | MAPE | 16.83 | 23.08 | 7.87 | 8.16 | 8.68 | **7.86** |
| | 12 | MAE | 1.03 | 1.39 | 0.64 | 0.64 | 0.75 | **0.63** |
| | | RMSE | 1.37 | 1.85 | 0.96 | 0.97 | 1.11 | **0.94** |
| | | MAPE | 16.83 | 22.63 | 8.38 | 8.35 | 9.03 | **8.30** |
| PEMSD7M | 3 | MAE | 4.01 | 3.45 | 2.29 | 2.37 | 2.88 | **2.28** |
| | | RMSE | 7.20 | 5.74 | 4.20 | 4.21 | 5.71 | **4.16** |
| | | MAPE | 10.61 | 9.07 | 5.04 | 5.54 | 7.25 | **3.93** |
| | 6 | MAE | 4.01 | 4.51 | 3.23 | 3.31 | **3.08** | 3.11 |
| | | RMSE | 7.20 | 7.44 | 5.74 | 5.96 | 6.17 | **5.72** |
| | | MAPE | 10.61 | 12.33 | 7.23 | 8.06 | 7.77 | **5.37** |
| | 12 | MAE | 4.01 | 6.18 | 4.45 | 4.31 | **3.99** | 4.03 |
| | | RMSE | 7.20 | 9.63 | 7.46 | 7.43 | 7.90 | **7.29** |
| | | MAPE | 10.61 | 17.61 | 7.67 | 10.29 | 10.02 | **6.94** |

#### 4.4.3. Ablation Experiments

We validate the effect of each component of the model on the prediction effect using three ablation experiments, where Model A indicates replacing the causal convolution with a normal two-dimensional convolution, and Model B indicates using the causal convolution and removing the temporal self-attention mechanism. Model C indicates using causal convolution and removing graph convolution. Table 2 shows the performance comparison of the ablation experiments.

According to Table 2, it can be seen that after replacing causal convolution with ordinary 2D convolution, the error metrics of multi-step prediction have increased, which indicates that compared with causal convolution, ordinary 2D convolution is unable to capture the temporal features of traffic flow. After Model B removes the temporal self-attention mechanism, the model is unable to identify important time points and subsequences in the traffic sequence, making the model prediction accuracy decrease. Graph convolution is more important to capture the spatial or structural dependence of the data, and the removal of graph convolution by Model C leads to a decrease in the model's ability to extract local features, which in turn affects the prediction accuracy.

Table 2. Results of ablation experiments.

| Dataset | Step | Metric | A | B | C | Ours |
|---|---|---|---|---|---|---|
| TrafficSH | 3 | MAE | 0.59 | 0.61 | 0.58 | 0.48 |
| | | RMSE | 0.89 | 0.98 | 0.89 | 0.78 |
| | | MAPE | 7.68 | 7.97 | 7.58 | 6.31 |
| | 6 | MAE | 0.68 | 0.77 | 0.76 | 0.60 |
| | | RMSE | 1.07 | 1.18 | 1.12 | 0.90 |
| | | MAPE | 8.94 | 10.15 | 9.97 | 7.86 |
| | 12 | MAE | 0.78 | 0.88 | 0.94 | 0.63 |
| | | RMSE | 1.14 | 1.30 | 1.34 | 0.94 |
| | | MAPE | 10.22 | 11.54 | 12.33 | 8.30 |
| PEMSD7M | 3 | MAE | 2.30 | 2.39 | 2.44 | 2.28 |
| | | RMSE | 4.19 | 4.37 | 4.30 | 4.16 |
| | | MAPE | 3.97 | 4.12 | 4.19 | 3.93 |
| | 6 | MAE | 3.15 | 3.32 | 3.14 | 3.11 |
| | | RMSE | 5.75 | 6.26 | 5.75 | 5.72 |
| | | MAPE | 5.42 | 5.72 | 5.41 | 5.37 |
| | 12 | MAE | 4.15 | 4.72 | 4.10 | 4.03 |
| | | RMSE | 7.34 | 8.45 | 7.31 | 7.29 |
| | | MAPE | 7.14 | 8.12 | 7.06 | 6.94 |

## 5. Conclusion

In this paper, a traffic flow prediction model based on the structure of autoencoder is proposed by combining CCN, GCN and MHSA. First, in the encoder, multiple 2D Causal Convolution modules are used to capture the core features of the traffic flow and pooling is employed to remove redundant information. Second, the attention weights are dynamically computed using a MHSA to identify important time points and sub-sequences in the traffic sequence, and the spatial features of the traffic flow captured by the GCN. Finally, when reconstructing the potential features in the decoder, the jump connection from the encoder is added, which makes the decoder reuse the shallow features extracted by the encoder in the feature reconstruction to get the prediction results. The experimental results show that compared with the baseline model, the model proposed in this paper is able to capture the spatio-temporal correlation of traffic speed data better in traffic flow prediction and has good prediction performance.

As mentioned in the Introduction, accurate traffic flow prediction is crucial for the efficient operation of intelligent transportation systems. The improved accuracy of this model can bring numerous tangible benefits to traffic management systems: it can help traffic management departments grasp the changing trends of traffic flow more timely, conduct early dredge on easily congested road sections, and reduce vehicle detention time; it can provide a more reliable basis for the optimization of traffic signal timing, improving road traffic efficiency; it can also offer accurate references for the public's travel route planning, enhancing travel experience.

In the future, we should not only use more datasets to verify the model validity, but also make further improvements to the model components to reduce the model complexity and improve the prediction accuracy. For example, wavelet decomposition can be adopted in data preprocessing to make the original data have both time-domain and frequency-domain features, and the processed data may lead to better results when input into the model.

## Acknowledgements

This research is supported in part by National Natural Science Foundation of China (NSFC) (Grant No.72461030).

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

[1] Okutani, I. and Stephanedes, Y.J. (1984) Dynamic Prediction of Traffic Volume through Kalman Filtering Theory. *Transportation Research Part B: Methodological*, **18**, 1-11. https://doi.org/10.1016/0191-2615(84)90002-x

[2] Liu, J. and Guan, W. (2004) A Summary of Traffic Flow Forecasting Methods. *Journal of Highway and Transportation Research and Development*, **21**, 82-85.

[3] Chorowski, J.K., Bahdanau, D. and Serdyuk, D. (2015) Attention-Based Models for

Speech Recognition. *Annual Conference on Neural Information Processing Systems* 2015, Montreal, 7-12 December 2015, 424-432.

[4] Williams, B.M. and Hoel, L.A. (2003) Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. *Journal of Transportation Engineering*, **129**, 664-672. https://doi.org/10.1061/(asce)0733-947x(2003)129:6(664)

[5] Cheng, S., Lu, F., Peng, P. and Wu, S. (2018) Short-Term Traffic Forecasting: An Adaptive ST-KNN Model That Considers Spatial Heterogeneity. *Computers, Environment and Urban Systems*, **71**, 186-198. https://doi.org/10.1016/j.compenvurbsys.2018.05.009

[6] Tang, J., Chen, X., Hu, Z., Zong, F., Han, C. and Li, L. (2019) Traffic Flow Prediction Based on Combination of Support Vector Machine and Data Denoising Schemes. *Physica A*: *Statistical Mechanics and Its Applications*, **534**, Article ID: 120642. https://doi.org/10.1016/j.physa.2019.03.007

[7] Liu, L., Chen, R., Zhao, Q. and Zhu, S. (2018) Applying a Multistage of Input Feature Combination to Random Forest for Improving MRT Passenger Flow Prediction. *Journal of Ambient Intelligence and Humanized Computing*, **10**, 4515-4532. https://doi.org/10.1007/s12652-018-1135-2

[8] Wang, Z. and Zhang, X.Q. (2010) Model of Road Traffic Accidents Prediction Based on ARIMA-FNN Optimal Weighted Combination. *Journal of Transport Information and Safety*, **28**, 89-92.

[9] Xie, P., Li, T., Liu, J., Du, S., Yang, X. and Zhang, J. (2020) Urban Flow Prediction from Spatiotemporal Data Using Machine Learning: A Survey. *Information Fusion*, **59**, 1-12. https://doi.org/10.1016/j.inffus.2020.01.002

[10] Chen, K., Liang, Y., Han, J., Feng, S., Zhu, M. and Yang, H. (2024) Semantic-Fused Multi-Granularity Cross-City Traffic Prediction. *Transportation Research Part C*: *Emerging Technologies*, **162**, Article ID: 104604. https://doi.org/10.1016/j.trc.2024.104604

[11] Gomes, B., Coelho, J. and Aidos, H. (2023) A Survey on Traffic Flow Prediction and Classification. *Intelligent Systems with Applications*, **20**, Article ID: 200268. https://doi.org/10.1016/j.iswa.2023.200268

[12] Cui, J.X., Yao, J. and Zhao, B.Y. (2024) Review on Short-Term Traffic Flow Prediction Methods Based on Deep Learning. *Journal of Traffic and Transportation Engineering*, **24**, 50-64.

[13] Li, Y., Gao, Y. and Yao, Z.X. (2025) Intelligent Traffic Flow Prediction for Data Scarcity Scenarios. *Journal of Software*, **36**, 3787-3801.

[14] Jiang, R., Wang, Z., Yong, J., Jeph, P., Chen, Q., Kobayashi, Y., *et al.* (2023) Spatio-Temporal Meta-Graph Learning for Traffic Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 8078-8086. https://doi.org/10.1609/aaai.v37i7.25976

[15] Yin, L., Liu, P., Wu, Y., Shi, C., Wei, X. and He, Y. (2023) ST-VGBiGRU: A Hybrid Model for Traffic Flow Prediction with Spatio-Temporal Multimodality. *IEEE Access*, **11**, 54968-54985. https://doi.org/10.1109/access.2023.3282323

[16] Yu, B., Yin, H.T. and Zhu, Z.X. (2018) Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Vol. 9, 3634-3640. https://doi.org/10.24963/ijcai.2018/505

[17] Ji, J., Wang, J., Huang, C., Wu, J., Xu, B., Wu, Z., *et al.* (2023) Spatio-Temporal Self-Supervised Learning for Traffic Flow Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 4356-4364.

https://doi.org/10.1609/aaai.v37i4.25555

[18] Qi, X., Yao, J., Wang, P., Shi, T., Zhang, Y. and Zhao, X. (2023) Combining Weather Factors to Predict Traffic Flow: A Spatial-Temporal Fusion Graph Convolutional Network-Based Deep Learning Approach. *IET Intelligent Transport Systems*, **18**, 528-539. https://doi.org/10.1049/itr2.12401

[19] Kong, W., Guo, Z. and Liu, Y. (2024) Spatio-Temporal Pivotal Graph Neural Networks for Traffic Flow Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, **38**, 8627-8635. https://doi.org/10.1609/aaai.v38i8.28707

[20] Jiang, J., Han, C., Zhao, W.X. and Wang, J. (2023) Pdformer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**, 4365-4373. https://doi.org/10.1609/aaai.v37i4.25556

[21] Ma, J., Zhao, J. and Hou, Y. (2024) Spatial-Temporal Transformer Networks for Traffic Flow Forecasting Using a Pre-Trained Language Model. *Sensors*, **24**, Article No. 5502. https://doi.org/10.3390/s24175502

[22] Shi, X.W., Li, L.C. and Yu, Z.X. (2025) Traffic Flow Prediction Based on Transformer with Spatio-Temporal Feature. *Journal of Chinese Computer Systems*, 1-9. https://link.cnki.net/urlid/21.1106.TP.20250425.0858.004

[23] Moon, H. and Cho, S. (2025) Traffic Prediction by Graph Transformer Embedded with Subgraphs. *Expert Systems with Applications*, **272**, Article ID: 126799. https://doi.org/10.1016/j.eswa.2025.126799

[24] Li, Y.G., Yu, R. and Cyrus, S. (2018) Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *International Conference on Learning Representations* 2018, Vancouver, 30 April-3 May 2018, 1-16.

[25] Zheng, C., Fan, X., Wang, C. and Qi, J. (2020) GMAN: A Graph Multi-Attention Network for Traffic Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 1234-1241. https://doi.org/10.1609/aaai.v34i01.5477

[26] Hu, Y.H., Wu, Q.M. and Zhu, D.L. (2009) Topological Properties and Vulnerability Analysis of Spatial Urban Street Networks. *Complex Systems and Complexity Science*, **6**, 69-76.

[27] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P. (2008) Extracting and Composing Robust Features with Denoising Autoencoders. *Proceedings of the* 25*th International Conference on Machine Learning*, Helsinki, 5-9 July 2008, 1096-1103. https://doi.org/10.1145/1390156.1390294

[28] Bai, S., Kolter, J.Z. and Koltun, V. (2018) An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *International Conference on Machine Learning*, Stockholm, 10-15 July 2018, 1-14. https://arxiv.org/abs/1803.01271

[29] Zhang, S., Zhang, J., Yang, L., Yin, J. and Gao, Z. (2023) Spatiotemporal Attention Fusion Network for Short-Term Passenger Flow Prediction on New Year's Day Holiday in Urban Rail Transit System. *IEEE Intelligent Transportation Systems Magazine*, **15**, 59-77. https://doi.org/10.1109/mits.2023.3265808

[30] Zhou, H.Q., Shi, B.Z. and Song, L.Y. (2025) Survey on Complex Spatio-Temporal Data Mining Methods Based on Graph Neural Network. *Journal of Software*, **36**, 1811-1843.