GADGN: A dual graph convolutional architecture for traffic flow prediction

Yanan Song[1], Siwei Wei[2] and Donghua Liu[3]

[1]*School of Computer Science, Hubei University of Technology, Wuhan, 430070, China*
[2]*CCCC Second Highway Consultants Co. Ltd., Wuhan, 430056, China*
[3]*Information Center, China Waterborne Transport Research Institute, Beijing, 100080, China*

Abstract

With the rapid growth of urban traffic, the importance of traffic flow prediction has increased. However, existing models still have significant limitations in capturing complex spatio-temporal dependencies: 1) The data collected by sensors is often incomplete, which can seriously affect the accuracy and reliability of traffic flow prediction. However, most existing models for processing missing values focus only on temporal dependencies, ignoring the complexity of spatio-temporal correlations in traffic data; 2) Relying only on predefined graph structures based on a priori knowledge or adaptive graph structures generated by node embedding often makes it difficult to accurately reflect the complexity and temporal variability of traffic flow data. However, most current models usually adopt only a single graph structure, fail to fully exploit the complementary advantages of multiple graph structures, and thus have certain limitations in capturing spatio-temporal features. To this end, this paper proposes a generative adversarial dual graph network for traffic flow prediction, viz, GADGN. In this paper, we use the GAN-TCT (Generative Adversarial Network-Transformer-CNN Fusion Module) module to achieve effective missing value filling. The generator of GAN-TCT combines a long short-term memory network (LSTM), a convolutional neural network (CNN) and a transformer to effectively extract the time-series features and spatial correlations in the input data, thus enhancing the realism of the generated data. In addition, this paper designs the dual graph convolutional fusion coding mechanism (DSGCE), which simultaneously considers the information contained in the node features and the adjacency matrix, generates an adaptive graph structure using node embeddings, constructs a predefined graph structure based on a priori knowledge through the adjacency matrix, and extracts the features contained in the two graph structures by convolution of the two graphs, and then extracts a more comprehensive graph structure from the two graph structures through the graph features learned by the self-encoder. It then extracts a more comprehensive graph structure from the two graph structures using the graph features learned by the self-encoder, and then extracts a more comprehensive spatio-temporal information from the two graph structures using the graph features learned by the self-encoder, and then combines this information to generate a new feature representation containing more potential patterns and dependencies, and finally captures the temporal features through a gating mechanism to obtain the final prediction results. Extensive experiments on real road datasets PeMSD4 and PeMSD8 have shown that the GADGN model reduces the MAE by about 3.5%, the RMSE by about 1.5% and the MAPE by about 1.8% compared to the existing state-of-the-art method (SOTA). These results fully demonstrate the superiority and effectiveness of the GADGN model in traffic flow prediction tasks.

Keywords

generative adversarial networks, dual graph convolutional fusion coding mechanism, missing value filling, gating mechanism, traffic flow prediction

1. Introduction

With the acceleration of urbanisation, the transport system is facing serious problems such as congestion and uneven public transport services. Traffic flow prediction has become a key issue in urban traffic management. Most of the traditional traffic flow prediction methods rely on cloud-based data centres for computation, but with the rapid development of sensor networks and the application of edge computing, more and more real-time data needs to be processed on edge devices. Edge computing, as a distributed computing architecture, can effectively reduce latency and improve data processing efficiency by distributing computing tasks to edge nodes close to the data source. In this context, this paper proposes a traffic flow prediction model (GADGN) based on generative adversarial network and

doors-2025: 5th Edge Computing Workshop, April 4, 2025, Zhytomyr, Ukraine
s1185401991@163.com (Y. Song); waosfengw@whut.edu.cn (S. Wei); liudonghua@wti.ac.cn (D. Liu)

dual graph convolutional architecture. The model combines the GAN-TCT module with the dual-graph convolutional fusion mechanism and is optimised for efficient data gap filling and traffic flow prediction in an edge computing environment.

Missing values in the data not only weaken the validity of the data, but also directly affect the reliability of model training, leading to inaccurate prediction results. Traditional filling methods, such as mean-filling, linear interpolation or forward filling, although simple and easy to use, often fail to capture the true distribution of the data. In addition, the relationships between nodes in urban traffic networks change dynamically, and traffic flows vary significantly depending on time, location and environment. Traditional models that rely only on static modelling are difficult to adapt to complex changes in traffic flows. Static neighbourhood matrices reflect the fixed connectivity of nodes in a transport network, and understanding these relationships is critical to capturing the underlying traffic patterns. Relying solely on dynamic modelling can result in models that do not fully capture changing traffic flow patterns. Therefore, the construction of effective traffic flow prediction models must focus on both static and dynamic spatio-temporal features to better adapt to complex traffic flow changes.

To solve these problems, this paper proposes a method that combines the GAN-TCT module with the dual graph convolutional fusion mechanism (GADGN) to improve the accuracy and robustness of traffic flow prediction. In the missing value filling session, we design the GAN-TCT module to efficiently generate pseudo-data that is highly similar to the real traffic flow to fill the missing parts of the data. The generator extracts time-series features related to missing values using a Long Short-Term Memory (LSTM) network, a Convolutional Neural Network (CNN) captures local spatial dependencies, and global context modelling is performed by the Transformer module. The structure not only integrates local and global features, but also effectively enhances the spatio-temporal consistency of the data through the feature fusion layer. In addition, this paper designs a dual-image convolutional fusion coding mechanism, which adopts a dual-image convolutional structure to fully capture the multi-scale features of both static and dynamic images, thus enhancing the feature extraction capability of the model. Specifically, static graph convolution uses a predefined graph structure to extract fixed neighbourhood features, while dynamic graph convolution dynamically generates time-varying neighbourhood matrices through node embedding, effectively characterising dynamic spatio-temporal dependencies. However, since static and dynamic graph convolution may only reflect local relationships when fused, and may not adequately model the deeper features in the data, we consider introducing an adaptive feature reconstruction module (self-encoder), which is capable of extracting complex spatio-temporal features through a hidden space representation, and helps to better represent the global and non-linear properties of the data. This mechanism effectively combines the a priori information of the fixed topology with the flexibility of the time-varying topology, enabling the model to better capture the long- and short-term dependencies and multi-scale features in the data when dealing with complex spatio-temporal data (e.g. traffic flow prediction), thus improving the overall prediction performance.).

2. Related work

2.1. Traffic flow forecasts

In the field of traffic flow prediction, researchers have developed a variety of deep learning models to improve prediction accuracy and capture complex spatio-temporal dependencies, each proposing innovative solutions to specific problems. For example, Graph WaveNet [1] significantly improves short-term prediction accuracy by combining the Graph Convolutional Network (GCN) and the Temporal Convolutional Network (TCN) to effectively capture local spatial dependencies and time-series patterns of traffic data. However, the model has limited ability to model dynamically changing spatio-temporal dependencies. AGCRN [2] improves the robustness of the model by introducing adaptive graph convolution and dynamically adjusting the relationships between nodes to adapt to different traffic conditions, but its reliance on dynamic graph structure may neglect the stabilising effect of fixed relationships. AST-GCN [3] combines graph convolution with a long- and short-term memory network (LSTM) to model the spatio-temporal dynamic properties in traffic data, which shows a good ability to capture temporal dependencies, but still falls short in processing complex spatial features. In addition, STHSGCN [4] and STFGCN [5] further improve the modelling ability of traffic flow data through hierarchical design and fusion of multiple spatial features, respectively. STHSGCN emphasises on capturing multi-level spatio-temporal dependencies, while STFGCN focuses on fusion of multi-dimensional spatial features to improve prediction accuracy. However, these models still have shortcomings in dealing with long-term dependencies and dynamic changes, especially in capturing and balancing fixed and time-varying relationships. The effective combination of fixed relationships, which contribute to robustness, and time-varying relationships, which better reflect real-time dynamics, remains a challenge.

To address the above issues, this paper proposes a Dual Graph Convolutional Fusion Coding (DSGCE) mechanism that comprehensively captures both fixed relationships and dynamically changing features in traffic data by combining an adaptive graph structure and a predefined adjacency matrix. The mechanism enhances the adaptability to dynamic changes in spatio-temporal dependence while maintaining the model's sensitivity to long-term stable features, providing a more comprehensive and efficient solution to the complex task of traffic flow prediction.

2.2. Missing value processing

In the field of missing value processing, traditional methods such as mean-filling and k-nearest neighbour algorithms [6] are widely used due to their ease of implementation, but such methods fail to capture the dynamic features in time series and are particularly inadequate for dealing with long-term missing values. Recent studies have shown that generative models have significant advantages in missing value filling. For example, generative models proposed by Dong et al. [7] and Yoon et al. [8] perform well in long-term interpolation tasks. However, these models typically treat the filling of missing values separately from the subsequent prediction, which can lead to information loss and a decrease in prediction accuracy. To overcome this limitation, researchers have attempted to use joint modelling approaches. For example, GRU-D [9] achieves the fusion of filling and prediction by directly estimating the missing values within the GRU, while LSTM-i [10] improves the modelling capability in the time dimension by using the LSTM state to infer the current missing values. However, these methods mainly focus on feature mining in the time dimension and fail to fully account for the complex spatio-temporal features and dynamic relationships between road segments in traffic data.

To address the above shortcomings, this paper proposes a Generative Adversarial Network-Transformer-CNN Fusion Module (GAN-TCT), which aims to comprehensively model the temporal variability characteristics and spatial dependence of traffic flow data. This multi-model fusion design effectively integrates local and global features, and achieves high robustness and accuracy in dynamic and complex traffic flow data.The GAN-TCT module not only provides an innovative method to fill in missing values, but also lays a solid foundation for improving the performance of traffic flow prediction.

The application of edge computing to traffic flow prediction is also gaining attention. In recent years, many studies have attempted to combine edge computing techniques with traffic flow prediction, aiming at real-time data processing and prediction using edge computing devices. For example, Yu et al. [11] proposed a short-term traffic flow prediction method based on spatio-temporal correlation using edge computing, which achieves fast traffic flow prediction and anomaly detection by deploying a deep learning model on the edge device, reducing the data transmission delay and improving the prediction accuracy. The GADGN model in this study precisely considers the application requirements of the edge computing environment, and by designing an efficient computing module adapted to the edge devices, it enables the model to better cope with the processing requirements of large spatio-temporal data in traffic flow prediction. In particular, the GADGN model proposed in this paper can operate efficiently in the edge computing environment. The GAN-TCT module of the model is able to process traffic flow data in real time on edge devices and generate high quality missing value filled data. The dual graph convolutional fusion coding mechanism is then able to extract spatio-temporal features through static and dynamic graph convolution and combine with a self-encoder for multi-scale feature learning. This design makes the model not only applicable to traditional cloud computing platforms, but also able to perform real-time traffic prediction and analysis on edge devices, improving real-time traffic management and decision support capabilities.

3. Modifications

The architecture of our proposed GADGN model is shown in figure 1, and consists of two main modules: the missing value filling module and the traffic prediction module. In the missing value filling module, we design the GAN-TCT module, which is different from the traditional GAN.The GAN-TCT module combines a Convolutional Neural Network (CNN) to extract local spatial features, a Long Short-Term Memory Network (LSTM) to capture time-series dependencies, and a Transformer module to model global spatio-temporal contextual information. Based on this structure, the Generative Adversarial Network (GAN) generates missing value filling results that are very close to the real data through adversarial training, thus achieving an important breakthrough in filling quality and spatio-temporal consistency, ensuring that the generator can simultaneously account for spatio-temporal dependencies and generate more realistic filling data. The filled data is divided into q time steps, and the data from each time step is used to generate a predefined knowledge-based adjacency matrix and a dynamic adjacency matrix based on node embeddings (without prior knowledge). These two adjacency matrices are then fed into two graph convolution modules to extract the spatial dependencies in the dynamic graph structure and the predefined knowledge-based graph structure, respectively. By combining the features from these two graph convolutions and feeding them into the self-encoder for feature learning, we are able to extract richer spatial feature information and complete comprehensive spatial feature extraction. Finally, by replacing the traditional update gate and reset gate with the extracted spatial features, we compute the gate control parameters to better capture the temporal dependencies in the data and obtain the final prediction results. The design not only improves the model's ability to model complex spatio-temporal dependencies, but also significantly improves the accuracy of the prediction.

3.1. Missing value processing module

Missing value handling is a key issue in spatio-temporal data analysis. This study proposes a new GAN-TCT module for missing value filling in traffic flow data. The module combines the generator and discriminator of Generative Adversarial Networks (GAN) while introducing a spatio-temporal feature extraction mechanism. The generator part consists of LSTM, CNN and Transformer architectures, which can effectively capture the temporal dependence and spatial relationship of the data. The discriminator part adopts MLP (Multi-Layer Perceptron), which discriminates the difference between the generated data and the real data through multiple fully connected layers to ensure that the generated data has high authenticity and consistency. The generator and discriminator optimise each other through adversarial training, improving the quality of the filling results. The model not only fills in missing values, but also captures more accurate spatio-temporal dependencies when dealing with complex spatio-temporal data, thus generating more realistic and accurate missing value filled data.

Specifically, the generator first captures long-term dependencies in the time series using LSTM, and then extracts local spatial features using CNN. The CNN-processed features are then fed into the Transformer encoder, which models the global spatio-temporal context information through the self-attention mechanism, further enhancing the ability to model complex spatio-temporal dependencies. To obtain global features, the generator also uses a global pooling layer that fuses the global representation of spatial features with the spatio-temporal features output by the Transformer to produce more realistic data filled with missing values. The generator is formulated as follows:

$$lstm = LSTM(x) \tag{1}$$

The LSTM process the output $lstm$, where x is the input data and $lstm$ is subjected to 1D convolution to extract local spatial features:

$$\mathbf{o}_{CNN} = \operatorname{Re} LU(CNN(lstm)) \tag{2}$$

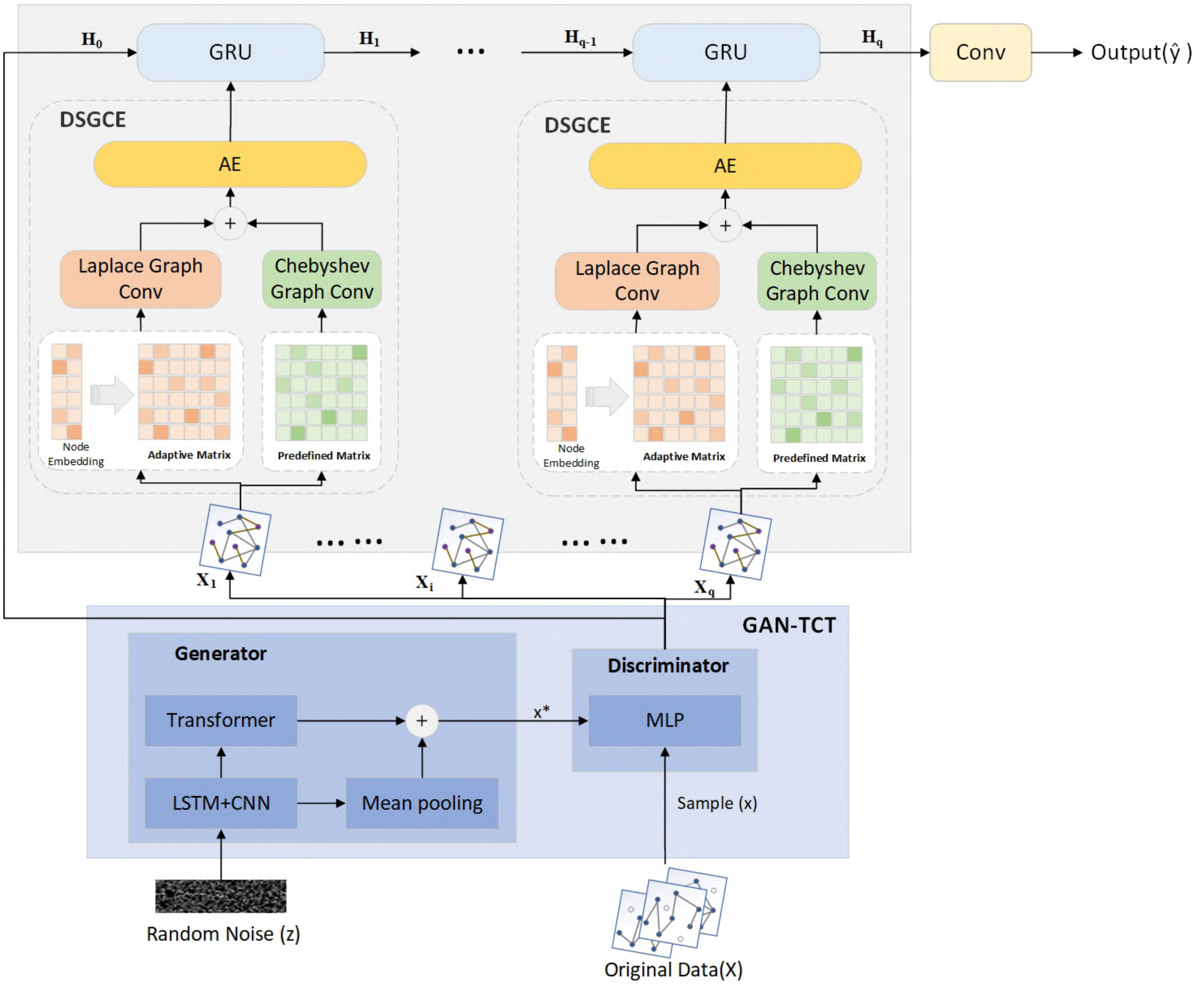

Figure 1: General framework of GADGN.

The features output from the convolutional layer are mapped to the input dimensions of the Transformer by a Fully Connected Layer (FC):

$$x_{trans} = FC(o_{CNN}) \tag{3}$$

The transformer input is processed by the encoder to extract global spatio-temporal features:

$$transformer = TransformerEncoder(x_{trans}) \tag{4}$$

To further enhance the features, the generator extracts global features from the output of the convolutional layer (by average pooling) and passes them to a fully connected layer:

$$C_{global} = FC\_g\left(mean(o_{CNN}, \mathrm{dim} = 1)\right) \tag{5}$$

Finally, the generator splices the features that come out of the transformer with the global features and fuses them together through a fully connected layer:

$$F_{fused} = cat(transformer, C_{global}, \mathbf{dim} = -1) \tag{6}$$

$$y_{gen} = FC\_f(F_{fused}) \tag{7}$$

where y_{gen} represents generated data. And the structure of the discriminator is based on several fully connected layers and its task is to distinguish between real and generated data. Its goal is to maximise the discrimination between real data y_{real} and generated data y_{gen}. The loss function of the discriminator is defined as:

$$L_D = -E\left[\log D(y_{real})\right] - E\left[\log(1 - D(y_{gen})\right] \tag{8}$$

where $D(\bullet)$ is the discriminator output.

In addition, we introduce the Kullback-Leibler Dispersion (KLD) as a measure of the difference between the distribution of the generated data and that of the real data:

$$KLD(P_{real}\|P_{gen}) = \sum_i P_{real}(i)\log\frac{P_{real}(i)}{P_{gen}(i)} \tag{9}$$

where P_{real} and P_{gen} represent the probability distribution of real and generated data respectively.

The total loss function of the generator is:

$$L_G = -E\left\lfloor \log D(y_{gen})\right\rfloor + \lambda \bullet KLD\left(P_{real}\|P_{gen}\right) \tag{10}$$

where the first term is the loss of the generator to fool the discriminator, the second term is the KLD between the distribution of the generated data and the distribution of the real data, and λ is the hyperparameter that controls the trade-off between the two.

During the training process, the generator and the discriminator improve each other through adversarial training, where the generator is continuously optimised to generate more realistic complementary missing value data, while the discriminator gradually improves its ability to discriminate between generated and real data. In addition, we design an adaptive data generation mechanism that enables GAN-TCT to dynamically adjust the generation process according to the spatio-temporal dependence of the data, in order to better capture patterns in complex traffic flow data and produce high-quality complementary results. This adaptive mechanism significantly reduces the filling error and improves the overall performance of the traffic flow prediction model.

3.2. Flow forecasting module

The traffic prediction module is the core part of this paper, and the accuracy of traffic prediction is crucial for traffic management. In order to improve the prediction accuracy, this paper proposes a dual graph convolutional fusion (DSGCE) coding mechanism, which not only effectively captures the spatial dependence in the data, but also dynamically adjusts the gating parameters by the spatial features of the self-encoder output to capture the temporal dependence to achieve more accurate traffic prediction. Specifically, this paper uses graph convolution based on the Laplace operator to capture the time-varying relationships in the dynamic neighbourhood matrix that reflect the instantaneous changes in traffic flow. Meanwhile, Chebyshev polynomial-based graph convolution is used to process the predefined adjacency matrix to model the stable relationships between nodes, thus capturing the long-term dependence of traffic flow. On this basis, this paper extracts and learns the spatial features using a self-encoder to obtain a feature representation with comprehensive spatial information. Based on the spatial features output from the self-encoder, the model dynamically adjusts the gating parameters to better capture the time dependence in the traffic flow data. By combining the spatial features obtained by the dual-image convolutional fusion coding mechanism with the temporal features adjusted by the gating mechanism, the model is able to more accurately predict the changes in the spatio-temporal characteristics of the traffic flow.

We divide the filled data combined with temporal information into q time segments, and the input data x_i of each time segment is the combination of the input data x_i of the time step and the output features H_{i-1} of the previous time step, and pass x_i to the Multilayer Perceptron (MLP) layer for feature extraction, which maps the learned dynamic features to the feature matrix P_i:

$$P_i = MLP(\mathrm{x}_i) \tag{11}$$

The obtained dynamic feature matrix P_i is combined with the node information matrix containing time information to generate the dynamic graph G_i^A for that time period. The first filled node embedding matrix A^N and the time information are subjected to an element-by-element product operation to obtain the node information matrix A_i^N with periodicity:

$$A_i^N = A^N \otimes T^d \otimes T^w \tag{12}$$

where T^d denotes the daily embedding and T^w denotes the weekly embedding. Secondly, the corresponding dynamic graph G_i^A is obtained by a further element-by-element multiplication operation of the dynamic feature matrix P_i and the periodic node matrix A_i^N for that time period:

$$\mathrm{G}_i^A = \tanh\left(P_i \otimes A_i^N\right) \tag{13}$$

Finally, the resulting dynamic graph is fed into the Laplace graph convolution for adaptive graph feature extraction:

$$\mathrm{D}_i^{-\frac{1}{2}}\mathrm{A}\mathrm{D}_i^{-\frac{1}{2}} = \mathrm{D}_i^{-\frac{1}{2}}\left(\operatorname{Re}\mathrm{LU}\left(\mathrm{G}_i^A\mathrm{G}_\mathrm{i}^{\mathrm{A}^T}\right)\right)\mathrm{D}_i^{-\frac{1}{2}} \tag{14}$$

where D_i is the diagonal matrix of time step i, whose diagonal elements are the degrees of each node. The dynamic graph convolution operation is performed by constructing a normalised Laplace matrix to generate the output feature V_i:

$$\mathrm{V}_i = \left(\mathrm{I}_N + \mathrm{D}_i^{-\frac{1}{2}}\mathrm{A}\mathrm{D}_i^{-\frac{1}{2}}\right)\mathrm{X}\theta + b \tag{15}$$

where $\mathbf{I}_N$ is the unit matrix, $\mathrm{X} \in \mathrm{R}^{N\times D}$ is the input feature for dynamic graph convolution, N represents the number of nodes of the matrix, D is the traffic feature captured by the sensor, and θ and b denote the weight and bias, respectively. And in this paper, the weight and bias matrices are also done an update operation through the node matrix to get the weight tensor $\mathbf{W}_i = A_i^N \bullet \theta$ and the bias term $b_i = A_i^N \bullet b$, which is obtained by combining the above two formulas:

$$\mathrm{V}_i = \left(\mathrm{I}_N + \mathrm{D}_i^{-\frac{1}{2}}\left(\operatorname{Re}\mathrm{LU}\left(G_i^A G_i^{A^T}\right)\right)\mathrm{D}_i^{-\frac{1}{2}}\right)\mathrm{X}\mathrm{W}_i + b_i \tag{16}$$

Similarly, our predefined-based adjacency matrix A_i^E is generated from the original adjacency matrix A^E combined with temporal information:

$$A_i^E = A^E \otimes T^d \otimes T^w \tag{17}$$

The generated pre-defined neighbourhood based matrix A_i^E is fed into the Chebyshev polynomial based graph convolution for feature extraction operation and finally outputs the pre-defined graph feature E_i:

$$Q_k = 2A_i^E Q_{k-1} - Q_{k-2}, k \geq 2 \tag{18}$$

$$\mathrm{E}_i = \sum_{k=0}^{U-1} \mathcal{Q}_k \mathrm{X}_E \tag{19}$$

where Q_0 and Q_1 are the initial polynomials, usually taken as $\mathcal{Q}_0 = I$ and $\mathcal{Q}_1 = A_i^E$, i.e. the unit matrix I and the adjacency matrix A_i^E of the predefined graph. In this paper, $k = 2$ and X_E are the input features of the Chebyshev graph convolution.

Next, the two graph features are combined and fed to the autocoder to further learn the spatial features and obtain a more comprehensive feature representation R_i. Specifically, the coder compresses the input features into the potential space Z via a two-layer linear transformation (weight matrices W_{enc1} and W_{enc2}) and an activation function, and the decoder starts from the potential space Z and reduces the feature matrices by using the two-layer linear transformation (weight matrices W_{dec1} and W_{dec2}) and activation functions to increase the similarity of the reconstructed features to the original input, reduces the feature matrix, and finally adds a residual join term $\mathrm{R}_{\mathrm{corv}}$ to increase the similarity of the reconstructed features to the original input:

$$\mathrm{R}_{\mathrm{conv}} = \mathrm{V}_i + \mathrm{E}_i \tag{20}$$

$$\mathrm{Z} = \sigma(W_{enc2} \cdot (\sigma(W_{enc1} \cdot \mathrm{R}_{\mathrm{conv}}))) \tag{21}$$

$$\mathrm{R}_i = \sigma(W_{dec2} \cdot (\sigma(W_{dec1} \cdot \mathrm{Z}))) + \mathrm{R}_{\mathrm{conv}} \tag{22}$$

Finally, we use the spatial features of the self-encoder output to compute gating parameters to dynamically extract temporal features from the data. The specific gating mechanism can be expressed as follows:

$$r_i = \sigma\left([x_i, \mathrm{H}_{i-1}]\, \mathrm{R}_i \mathrm{W}_r + b_r\right) \tag{23}$$

$$z_i = \sigma\left([x_i, \mathrm{H}_{i-1}]\, \mathrm{R}_i \mathrm{W}_z + b_z\right) \tag{24}$$

$$c_i = \tanh\left([x_i, \mathrm{H}_{i-1} \odot r_i]\, \mathrm{R}_i \mathrm{W}_c + b_c\right) \tag{25}$$

$$\mathrm{H}_i = z_i \odot \mathrm{H}_{i-1} + (1 - z_i) \odot c_i \tag{26}$$

where σ denotes sigmoid activation, W_r , W_z , W_c , b_r , b_z , b_c denote learnable parameters, [•] denotes splicing operation, R_i denotes reset and update gates, c_i is an intermediate temporary variable, and H_i denotes the spatio-temporal characteristics of the output of the gating mechanism at the i-th time step.

The spatio-temporal dependent information captured at each time step is output through the gating mechanism. We take the gated feature output of the q-th time step as the final result, i.e., H_q , and further process it through the convolutional layer to produce the final prediction result $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = conv(\mathrm{H}_q) \tag{27}$$

3.3. Loss function

The loss function plays a crucial role in the training process and is used to evaluate the performance of the model during the process. By minimising the loss function, the model can appropriately reduce the difference between the predicted values and the true labels, thus improving the accuracy of the prediction. We use L1 loss as the loss function to train the whole model. Where K is the historical time step, T is the future time step and L_s is the absolute value of the difference between the true value Y_{K+i} and the predicted value $\hat{\mathrm{Y}}_{K+i}$:

$$L_s = \frac{1}{T} \sum_{i=1}^{T} \left| \hat{\mathrm{Y}}_{K+i} - \mathrm{Y}_{K+i} \right| \tag{28}$$

Finally, we combine the total loss function L_G of the generator with the loss values L_s trained on the whole model to obtain the total loss:

$$L = \gamma L_G + L_s \tag{29}$$

where γ is a weighting parameter to balance the contribution of the two loss components.

4. Experimental setup

4.1. Datasets

To evaluate the performance of the GADGN proposed in this paper, we use two publicly available real traffic flow datasets from the PeMS data: the PeMSD4 and PeMSD8. The PeMS data are collected in real time from more than 40,000 detectors in California, USA.

Specific dataset information is provided below:

- PeMSD4: This dataset consists of 307 detectors collecting data at 5-minute intervals for a total of 59 days of traffic flow data. The data was collected between January and February 2018.
- PeMSD8: This traffic dataset consists of 170 detectors collecting data at 5-minute intervals for a total of 62 days of data, collected between July and August 2016.

Each collection of both datasets contains features in three dimensions: volume, average speed and average occupancy.

4.2. Experimental setup

We ran our experiments on a Win10 PC with an NVIDIA GeForce RTX 2070 graphics card. Specifically, our data are divided in a ratio of 6:2:2, where 60% of the data set is used as the training set, 20% of the data set is used as the validation set, and the remaining 20% is the test set. Regarding the hyperparameters of the GADGN model, we set the hidden size to 64, the kth-order Chebyshev polynomial expansion to 2, the embedding dimension D of PEMSD4 to 10, and the embedding dimension D of PEMSD8 to 5. The model was trained using the Adam optimiser and the training time was set to 100. Due to the combination of several modules such as biplot convolution, generative adversarial and self-encoder, the model has a high memory requirement, the batch size of the PEMSD4 dataset was set to 8 and the learning rate to 0.00075 for the PEMSD4 dataset, and the batch size was set to 24 and the learning rate to 0.0015 for the PEMSD8 dataset. The MAE was used as the loss function for training the model. Three evaluation metrics were used to assess performance: (1) mean absolute error (MAE), (2) root mean square error (RMSE) and (3) mean absolute percentage error (MAPE).

(1) Mean absolute error:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\mathrm{Y}_{K+i} - \hat{\mathrm{Y}}_{K+i}\right| \tag{30}$$

(2) Root mean square error:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{Y}_{K+i} - \hat{\mathrm{Y}}_{K+i}\right)^2} \tag{31}$$

(3) Mean absolute percentage error:

$$MAPE = \frac{100\%}{N}\sum_{\mathrm{i}=1}^{N}\left|\frac{\widehat{\mathrm{Y}}_{K+i} - \mathrm{Y}_{K+i}}{\mathrm{Y}_{K+i}}\right| \tag{32}$$

4.3. Baseline

We compare the proposed GADGN with the following 10 representative traffic forecasting methods:

- ARIMA [12]: combines three techniques, namely autoregressive (AR), differential (I) and moving average (MA), which can effectively capture trends and cyclical changes in data by modelling time series for future data forecasting.
- VAR [13]: captures the dynamic interactions between individual variables by using the lagged values of multiple time series as explanatory variables. It is a multivariate linear model widely used in time series analysis and forecasting, and is particularly suitable for dealing with the interdependence between multiple time series variables.
- STGCN [14]: it is a model that combines Graph Convolutional Networks (GCNs) and Convolutional Neural Networks (CNNs), specifically designed to deal with complex spatial and temporal dependencies in spatio-temporal data.

- ASTGCN [3]: combines the ideas of Graph Convolutional Networks (GCNs) and attention mechanisms to capture spatial dependence through GCNs, enhance time-dependent modelling through attention mechanisms, and adaptively select important features in spatio-temporal dimensions to improve modelling and prediction of complex spatio-temporal data.
- STSGCN [15]: it is a deep learning approach to modelling spatio-temporal data that aims to capture the complex correlations and dynamic evolution between spatio-temporal data by processing spatial and temporal dimensions simultaneously. The model models the relationships between data in the spatio-temporal dimension by combining graph convolution and temporal convolution, and by processing spatial and temporal features synchronously.
- Graph WaveNet [1]: combines the ideas of graph neural networks and WaveNet models. Efficient modelling of complex spatio-temporal data is achieved by effectively fusing spatial and temporal information, using graph neural networks to capture spatial dependencies in the data and WaveNet to capture temporal dependencies.
- STFGNN [16]: it is a deep learning model based on Graph Neural Network (GNN) architecture that aims to effectively capture and fuse spatial and temporal features in spatio-temporal data. This model is able to integrate spatial and temporal dependencies in graph-structured data by introducing a spatio-temporal fusion mechanism.
- AGCRN [2]: its main innovation is to adaptively learn the spatial relationships between nodes in a traffic network by introducing an attention mechanism, combining the advantages of graph convolution and recurrent neural networks, which significantly improves the model's performance and prediction accuracy in spatio-temporal data modelling.
- STHSGCN [4]: it is a novel spatio-temporal graph convolutional network model designed to address the shortcomings of traditional spatio-temporal graph neural networks in modelling spatial and temporal heterogeneity. The model is able to capture heterogeneity and causality in spatio-temporal data more accurately by designing a separate extended causal spatio-temporal synchronisation graph convolutional network and combining it with a causal spatio-temporal synchronisation graph mechanism.
- STFGCN [5]: combines the advantages of graph convolution and time series learning to propose a new fusion mechanism capable of handling complex traffic flow data. Special emphasis is placed on learning the spatial dependence and temporal correlation in traffic sequences, while the model's ability to model spatio-temporal data is further enhanced by the introduction of a continuous time correlation learning module and a transformer-based global time correlation learning module.

Table 1
Comparison of GADGN with baseline.

Model	PEMSD4			PEMSD8		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	33.73	48.80	24.18%	31.09	44.32	22.73%
VAR	24.54	38.61	17.24%	19.19	29.81	13.10%
STGCN	23.64	36.43	14.70%	19.00	28.70	11.32%
ASTGCN	21.38	33.83	14.18%	18.25	28.06	11.64%
STSGCN	21.19	33.65	13.90%	17.13	26.80	10.96%
Graph WaveNet	24.89	39.66	17.29%	18.28	30.05	12.15%
STFGNN	20.48	32.51	16.77%	16.94	26.25	10.60%
AGCRN	19.83	32.26	12.97%	15.95	25.22	10.09%
STHSGCN	19.50	31.39	12.89%	15.50	24.51	9.95%
STFGCN	18.95	30.90	12.36%	15.23	24.35	9.83%
GADGN	18.29	30.44	12.14%	14.27	23.67	9.38%

A comparison of the results of our model (GADGN) with the other 10 benchmark models on the PeMSD4 and PeMSD8 datasets is shown in table 1. All models are experimented on the same dataset and their performance is evaluated over 12 time steps. It can be seen that GADGN shows excellent performance in all three metrics, MAE, RMSE and MAPE, outperforming all the benchmark models.

- Comparison with traditional methods: Traditional time series models such as ARIMA and VAR have significant advantages in modelling temporal correlation. However, they are limited in that they do not account for spatial heterogeneity in the transport network, making it difficult to deal with spatial dependencies in complex transport systems. This leads to poor performance in modelling non-linear features, especially on the PEMS04 and PEMS08 datasets, which have a large number of nodes and high data complexity, and significantly poor prediction performance. In contrast, our model GADGN is able to comprehensively capture spatio-temporal features by introducing a spatio-temporal dependency-based missing value filling method and a dual-plot convolutional fusion coding mechanism, while retaining the ability to model temporal dependencies, which significantly improves the prediction accuracy and stability.
- Comparison with fixed neighbourhood matrix: Models such as STGCN, which are based on a fixed adjacency matrix, are able to capture spatial dependencies in traffic flows using a predefined spatial structure. However, the limitation of such models lies in the static nature of the adjacency matrix, which is unable to adapt to the dynamically changing characteristics of the traffic network and thus lacks flexibility in dealing with complex spatio-temporal relationships, limiting their predictive performance. Our GADGN model innovatively combines an adaptive adjacency matrix and a self-encoder, which can not only flexibly capture the static spatial relationships, but also dynamically adjust the spatial dependencies according to different time steps, comprehensively expressing the global and non-linear characteristics of the data, thus improving the sensitivity to traffic flow changes and the prediction accuracy.
- Compared to Adaptive Neighborhood Matrix: Models such as Graph WaveNet and AGCRN are able to dynamically adjust the adjacency matrix by learning the spatial correlations in the data, thus outperforming models based on fixed adjacency matrices in capturing dynamic changes in the traffic network. However, these models rely too heavily on adaptive adjacency matrices, often ignoring the a priori knowledge already present in the traffic network and making it difficult to fully exploit fixed spatial relationships. Our GADGN model innovatively combines the adaptive adjacency matrix with a predefined graph structure and uses the Dual Graph Convolutional Fusion Coding (DSGCE) mechanism to achieve efficient fusion of static and dynamic spatial relationships. Through this mechanism, the model not only flexibly adjusts the adjacency matrix to adapt to dynamic changes, but also fully exploits the a priori knowledge to comprehensively capture the multi-scale spatio-temporal features in the traffic flow, which further improves the prediction performance.

To more intuitively demonstrate the performance difference between GADGN and other models, we visualise the above tabular data as shown in figure 2.

4.4. Ablation experiments

To evaluate the performance of our proposed GADGN model, we conducted a series of ablation experiments using the PEMS04 and PEMS08 datasets. By removing different components of the model, we analysed the contribution of each module to the overall performance of the model with the following experimental design:

- w/o Trans: Removal of the Transformer module from the GAN-TCT framework to assess its impact on the accuracy of missing value filling.
- w/o CNN: Removal of Convolutional Neural Networks (CNNs) from the GAN-TCT module to evaluate the importance of CNNs in capturing spatially dependent features during missing value filling.
- w/o PG: Removing the neighbourhood matrix based on predefined knowledge and using only the dynamic neighbourhood matrix generated based on node embedding to assess the importance of a priori knowledge in the construction of the graph structure.

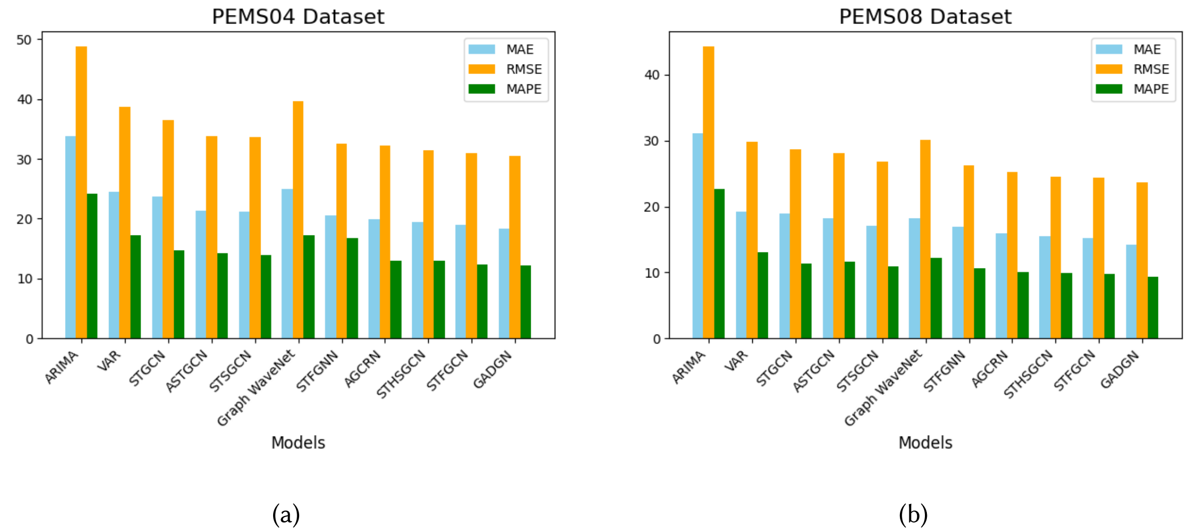

Figure 2: Visualisation of GADGN performance against other models on two datasets, (a) and (b).

- w/o AE: The self-encoder module is removed and the output features of the biplot convolution are passed directly to the following module without going through the learning and fusion process of the self-encoder.

The results of the experiments are summarised in table 2. As can be seen from the table, the results of the ablation experiments are inferior to the GADGN model for both the PEMSD4 and PEMSD8 datasets. In particular, the results of the w/o Trans experiments show that the removal of Transformer significantly reduces the model's ability to capture global spatio-temporal dependent features, which further validates the important role of Transformer in the missing value filling task for global feature extraction. The w/o CNN experiments show that the removal of Convolutional Neural Network (CNN) significantly reduces the model's ability to capture global spatio-temporal dependent features, which fully illustrates the key role of CNN in learning spatial dependencies in the GAN-TCT module. The w/o PG experiments show that using only adaptive adjacency matrices significantly weakens the model's ability to capture complex spatio-temporal dependencies, leading to a decrease in prediction accuracy, highlighting the importance of predefined knowledge-based graph structures in modelling complex spatial dependencies. Finally, the w/o AE experimental results show that the removal of the self-encoder mechanism significantly degrades the model performance, further confirming the indispensability of the self-encoder in fusing and reconstructing graph convolutional features, which effectively enhances the model's ability to capture and express spatial dependency features. In summary, the design and implementation of the individual components contribute significantly to the overall performance of the GADGN model, further demonstrating its effectiveness in spatio-temporal data prediction tasks.

5. Conclusion

The GADGN model proposed in this paper significantly improves the missing value filling and prediction ability of spatio-temporal data by introducing the GAN-TCT module and the dual graph convolutional fusion encoding mechanism (DSGCE). The missing value filling module adopts the Generative Adversarial Network (GAN), which combines the Long Short Term Memory Network (LSTM), Convolutional Neural Network (CNN) and Transformer to accurately capture spatio-temporal dependencies and produce filling results that are closer to the real data. The feature extraction module effectively integrates static and dynamic adjacency matrices through the synergistic action of the dual-image convolutional fusion coding mechanism and the gating mechanism, which not only strengthens the learning ability of complex spatial dependencies, but also dynamically captures temporal dependencies, thus improving

Table 2
Experimental results of variants of GADGN on both databases.

Method (PEMSD4)	MAE	RMSE	MAPE
GADGN	18.29	30.44	12.14%
w/o Trans	18.63	31.01	12.38%
w/o CNN	18.56	30.82	12.36%
w/o PG	18.54	30.71	12.48%
w/o AE	18.54	30.77	12.52%
Method (PEMSD8)	**MAE**	**RMSE**	**MAPE**
GADGN	14.27	23.67	9.38%
w/o Trans	14.59	24.08	9.58%
w/o CNN	14.43	23.75	9.45%
w/o PG	14.34	23.77	9.41%
w/o AE	14.37	23.92	9.52%

the prediction performance of the model.

Despite the excellent results achieved by the GADGN model in centralised computing architectures, its application in edge computing environments still faces challenges in terms of computational resources and real-time performance, especially in terms of inference speed and computational efficiency, which need to be further optimised. In addition, although the dual-image convolutional fusion coding mechanism can effectively capture spatio-temporal dependencies, the model may have difficulty dealing with anomalies caused by extreme events such as unexpected accidents or severe weather in certain complex traffic networks, which may affect the accuracy of the prediction. Future research will focus on improving the computational efficiency and inference speed of the model on edge devices, and consider the introduction of external influences to promote widespread adaptation and extension of the model in practical applications.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China under Grants 62206116 and Chunhui Plan Collaborative Research Project, Ministry of Education, China (HZKY20220350)& Key R&D Plan of Hubei Provincial Department of Science and Technology 2023BCB041.

Declaration on Generative AI: The authors have not employed any generative AI tools.

References

[1] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph WaveNet for Deep Spatial-Temporal Graph Modeling, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 1907–1913. doi:10.24963/ijcai.2019/264.

[2] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/ce1aad92b939420fc17005e5461e6f48-Abstract.html.

[3] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 922–929. doi:10.1609/aaai.v33i01.3301922.

[4] X. Yu, Y.-X. Bao, Q. Shi, STHSGCN: Spatial-temporal heterogeneous and synchronous graph convolution network for traffic flow prediction, Heliyon 9 (2023) e19927. doi:10.1016/j.heliyon.2023.e19927.

[5] Y. Ma, H. Lou, M. Yan, F. Sun, G. Li, Spatio-temporal fusion graph convolutional network for traffic flow forecasting, Information Fusion 104 (2024) 102196. doi:10.1016/j.inffus.2023.102196.

[6] G. E. A. P. A. Batista, M. C. Monard, A Study of K-Nearest Neighbour as an Imputation Method, in: A. Abraham, J. Ruiz-del-Solar, M. Köppen (Eds.), Soft Computing Systems - Design, Management and Applications, HIS 2002, December 1-4, 2002, Santiago, Chile, volume 87 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2002, pp. 251–260.
[7] H. Dong, F. Ding, H. Tan, H. Zhang, Laplacian integration of graph convolutional network with tensor completion for traffic prediction with missing data in inter-city highway network, Physica A: Statistical Mechanics and its Applications 586 (2022) 126474. doi:10.1016/j.physa.2021.126474.
[8] J. Yoon, D. Jarrett, M. van der Schaar, Time-series Generative Adversarial Networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.
[9] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent Neural Networks for Multivariate Time Series with Missing Values, Scientific Reports 8 (2018) 6085. doi:10.1038/s41598-018-24271-9.
[10] Z. Cui, R. Ke, Z. Pu, Y. Wang, Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values, Transportation Research Part C: Emerging Technologies 118 (2020) 102674. doi:10.1016/j.trc.2020.102674.
[11] X. Yu, L. Sun, Y. Yan, G. Liu, A short-term traffic flow prediction method based on spatial–temporal correlation using edge computing, Computers & Electrical Engineering 93 (2021) 107219. doi:10.1016/j.compeleceng.2021.107219.
[12] B. M. Williams, L. A. Hoel, Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results, Journal of Transportation Engineering 129 (2003) 664–672. doi:10.1061/(ASCE)0733-947X(2003)129:6(664).
[13] E. Zivot, J. Wang, Vector Autoregressive Models for Multivariate Time Series, in: Modeling Financial Time Series with S-PLUS®, Springer New York, New York, NY, 2006, pp. 385–429. doi:10.1007/978-0-387-32348-0_11.
[14] B. Yu, H. Yin, Z. Zhu, Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, in: J. Lang (Ed.), Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 3634–3640. doi:10.24963/IJCAI.2018/505.
[15] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 914–921. doi:10.1609/aaai.v34i01.5438.
[16] M. Li, Z. Zhu, Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting, Proceedings of the AAAI Conference on Artificial Intelligence 35 (2021) 4189–4196. doi:10.1609/aaai.v35i5.16542.