蘭州理工大學

科研成果汇总

学　　号: 181081103001

研 究 生: 肖雯娟

导　　师: 王晓明 教授

研究方向: 智能交通系统

论文题目: 基于数据驱动的交通流预测与交通状态判别方法研究

学　　科: 系统工程

学　　院: 电气工程与信息工程学院

入学时间: 2018 年 9 月

2024 年 11 月 4 日

目 录

图书馆 文献检索报告

兰州理工大学图书馆 LUTLIB 报告编号：R2024-1056

机构：兰州理工大学

姓名：肖雯娟 [181081103001]

著者要求对其在国内外学术出版物所发表的科技论著被以下数据库收录情况进行查证。

检索范围：

- 科学引文索引（Science Citation Index Expanded）： 1900年-2024年
- 工程索引（Engineering Index）： 1884年-2024年

检索结果：

检索类型	数据库	年份范围	总篇数	第一作者篇数
SCI-E 收录	SCI-EXPANDED	2023	1	1
EI 收录	EI-Compendex	2023 - 2024	3	3

End

委托人声明：

本人委托兰州理工大学图书馆查询论著被指定检索工具收录情况，经核对检索结果，附件中所列文献均为本人论著，特此声明。

作　者（签字）：肖雯娟

完成人（签字）：陈超

完　成　日　期　：2024年10月31日

完成单位（盖章）：兰州理工大学图书馆信息咨询与学科服务部

（本检索报告仅限校内使用）

图书馆

文献检索报告
SCI-E 收录

兰州理工大学图书馆 LUTLIB

报告编号：**R2024-1056 SCI-E** 收录

数据库：科学引文索引 **(Science Citation Index Expanded)** 时间范围：**2023**年	作者姓名：肖雯娟 作者单位：兰州理工大学	检索人员：陈超 检索日期：**2024**年**10**月**31**日

检索结果：被 **SCI-E** 收录文献 **1** 篇

#	作者	地址	标题	来源出版物	文献类型	入藏号
1	**Xiao, WJ**; Wang, XM	[Xiao, Wenjuan; Wang, Xiaoming] Lanzhou Univ Technol, Coll Elect & Informat Engn, Lanzhou 730050, Peoples R China.; [Xiao, Wenjuan] Northwest Minzu Univ, Coll Elect Engn, Lanzhou 730030, Peoples R China.	Spatial-Temporal Dynamic Graph Convolutional Neural Network for Traffic Prediction	***IEEE ACCESS*** 2023, 11: 97920-97929.	J Article	WOS:001068792500001
					合计	1

图书馆

文献检索报告
EI 收录

SEARCH

兰州理工大学图书馆 LUTLIB 报告编号：**R2024-1056 EI 收录**

数据库：工程索引 **(Engineering Index)** 时间范围：**2023年至2024年**	作者姓名：肖雯娟 作者单位：兰州理工大学	检索人员：陈超 检索日期：**2024年10月31日**

检索结果：被 **EI** 收录文献 **3** 篇

#	作者	地址	标题	来源出版物	文献类型	入藏号
1	**Xiao, Wenjuan**; Wang, Xiaoming	College of Electrical and Information Engineering, Lanzhou University of Technology, Gansu, Lanzhou; College of Electrical Engineering, Northwest Minzu University, Gansu, Lanzhou	Attention Mechanism Based Spatial-Temporal Graph Convolution Network for Traffic Prediction	***Journal of Computers (Taiwan)*** 2024, 35 (4): 93-108.	Journal article (JA)	20243817056240
2	**Xiao, Wenjuan**; Wang, Xiaoming	Lanzhou University of Technology, College of Electrical and Information Engineering, Lanzhou; Northwest Minzu University, College of Electrical Engineering, Lanzhou	Spatial-Temporal Dynamic Graph Convolutional Neural Network for Traffic Prediction (Open Access)	***IEEE Access*** 2023, 11: 97920-97929.	Journal article (JA)	20233814762618
3	**Xiao, Wenjuan**; Lv, Hui	College of Electrical and Information Engineering, Lanzhou University of Technology, China; College of Electrical Engineering, Northwest MinZu University, China	Research on Time Series Imputation Based on Generative Adversarial Network (Open Access)	***Advances in Transdisciplinary Engineering*** 2023, 35: 428-436.	Conference article (CA)	20234114859119
					合计	3

Journal of Computers Vol. 35 No. 4, August 2024, pp. 93-108
doi: 10.53106/199115992024083504007

Attention Mechanism Based Spatial-Temporal Graph Convolution Network for Traffic Prediction

Wenjuan Xiao[1,2*] and Xiaoming Wang[1]

[1] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, Gansu, China

xwj@xbmu.edu.cn

[2] College of Electrical Engineering, Northwest Minzu University, Lanzhou 730030, Gansu, China

Received 23 October 2023; Revised 30 January 2024; Accepted 14 March 2024

Abstract. Considering the complexity of traffic systems and the challenges brought by various factors in traffic prediction, we propose a spatial-temporal graph convolutional neural network based on attention mechanism (AMSTGCN) to adapt to these dynamic changes and improve prediction accuracy. The model combines the spatial feature extraction capability of graph attention network (GAT) and the dynamic correlation learning capability of attention mechanism. By introducing the attention mechanism, the network can adaptively focus on the dependencies between different time steps and different nodes, effectively mining the dynamic spatial-temporal relationships in the traffic data. Specifically, we adopt an improved version of graph attention network (GAT_v2) in the spatial dimension, which allows the model to capture more complex dynamic spatial correlations. Furthermore, in the temporal dimension, we combine gated recurrent unit (GRU) structure with an attention mechanism to enhance the model's ability to process sequential data and predict traffic flow changes over prolonged periods. To validate the effectiveness of the proposed method, extensive experiments were conducted on public traffic datasets, where AMSTGCN was compared with five different benchmark models. Experimental results demonstrate that AMSTGCN exhibits superior performance on both short-term and long-term prediction tasks and outperforms other models on multiple evaluation metrics, validating its potential and practical value in the field of traffic prediction.

Keywords: transportation system, attention mechanism, dynamic change, spatial-temporal dependency

1 Introduction

With the acceleration of urbanization, traffic congestion has become a common phenomenon in modern life. Accurate prediction of traffic conditions plays a very important role in improving the efficiency of transportation systems, reducing congestion, optimizing route selection, providing real-time navigation, and planning urban development.

As a key link in urban management and smart transportation systems, traffic prediction is becoming increasingly important. However, there are still many challenges in this field, in particular the need to accurately capture dynamically changing spatial correlations and complex temporal dependencies [1]. These factors significantly increase the complexity and difficulty of the prediction. To help the reader understand these challenges intuitively, we use Fig. 1 and Fig. 2 to provide specific examples for illustration. With these graphs, we show the spatial-temporal dynamics of traffic flows and the difficulty that traditional models have in capturing such dynamics.

(1) Dynamic Spatial Correlations. In previous studies, the spatial correlations between nodes are commonly represented by predefined static adjacency matrices, as mentioned in reference [2]. However, in real traffic environments, the spatial relationships between roads are dynamic systems subject to various factors such as traffic accidents and traffic regulations. Fig. 1 illustrates a schematic diagram of the road network in a certain urban area, where A, B, C, and D represent four intersections equipped with traffic detectors and treated as nodes in the network. For example, if the traffic authority implements a rule at A intersection that prohibits left turns from east to west, this will directly affect the traffic flow relationship between points A and B. Specifically, due to this restriction, the direct traffic flow from point A to point B will decrease, which means that the impact of traffic volume at point A on point B will correspondingly decrease, while the impact of point B on point A will rela-

* Corresponding Author

tively increase. Additionally, although point C and point B are not directly connected geographically, due to the left-turn restriction at point A, vehicles wishing to travel from east to west to reach point B must detour through points C and D, indirectly affecting the traffic conditions at point B. Furthermore, once the traffic rules at point A change, the spatial correlations of the entire A, B, C, and D node network will also adjust accordingly. With such an analysis, we can identify the dynamic nature of spatial relationships in real traffic scenarios and their impact on the design of predictive models.

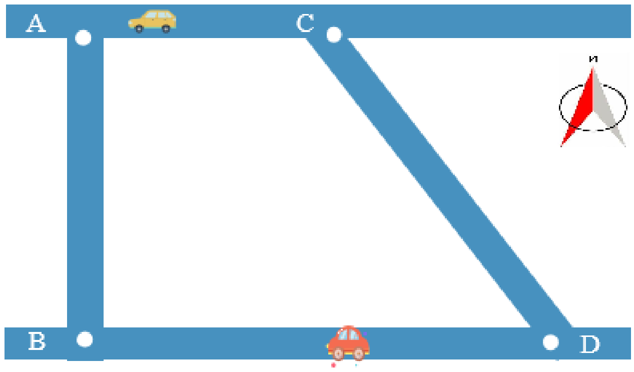


Fig. 1. Dynamic spatial-temporal relationship diagram of road node traffic flow

(2) Complex and Variable Temporal Dependencies. Traffic data inherently exhibit unique features on different time scales, including hours, days, weeks, and seasons. For example, morning and evening rush-hour traffic flows experience significant fluctuations, and weekday congestion patterns can be very different from those on weekends. In addition, temporal dependencies of traffic flow data can shift due to external events such as traffic accidents, weather fluctuations, or public gatherings. Fig. 2 illustrates the daily periodicity of traffic volume but also reveals an anomaly: a significant decrease in traffic volume during the period from 14:50 to 16:40 on Monday afternoons. The anomaly could be caused by congestion caused by factors such as road accidents or road construction. Therefore, when constructing traffic prediction models, it is necessary not only to take into account the regular temporal evolution of traffic data but also to capture and model temporal dependencies at different time scales, as well as potential anomalies that may arise. Such a comprehensive consideration is crucial for improving prediction accuracy.

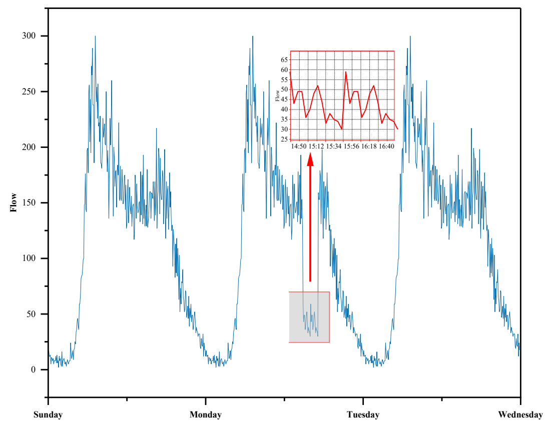


Fig. 2. The daily periodicity and dynamic variations of traffic flow volume

Through in-depth analysis, we have gained a new understanding of the importance of dynamic spatial-temporal relationships in traffic data. These relations include not only the periodic variations of traffic flow over time, but also sudden changes caused by unpredictable factors such as traffic accidents, weather conditions, and temporary road closures. Moreover, spatial correlations are continuously affected by factors such as traffic regulations, road network structure, and urban planning. These dynamic spatial-temporal dependencies require traffic prediction models to have a high level of adaptability and flexibility to accurately capture and respond to these complex traffic patterns.

In traditional traffic prediction methods, statistical models such as Autoregressive Integrated Moving Average (ARIMA), Exponential Smoothing models, and Regression models have dominated. However, these models have significant limitations when dealing with complex, nonlinear, and high-dimensional traffic data. They typically rely on manually extracted features and are built on linear assumptions, which limits their ability to capture the deep complexity of traffic patterns. With the introduction of machine learning, especially deep learning models, the field of traffic prediction has made significant advances. Deep learning models such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) [3] have shown superiority in multiple application scenarios by automatically learning features from the data. However, these models generally require a large amount of training data and have room for improvement in modelling the dynamics of spatial-temporal data.

To address these issues, we propose a spatial-temporal graph convolutional neural network (AMSTGCN) that incorporates an attention mechanism. This model not only automatically learns spatial-temporal features from complex traffic data, but also effectively adapts to spatial-temporal relationships with different time scales and dynamic changes. Through experimental validation on public datasets, AMSTGCN demonstrated its superiority in short-term and long-term traffic prediction tasks, demonstrating its effectiveness in capturing and predicting dynamic spatial-temporal traffic data. The main contributions of this study are as follows:

(1) Spatial-temporal relationship modelling. We propose an improved graph attention network (GAT_v2) [4] approach that dynamically extracts spatial relationships instead of relying on a static adjacency matrix used in traditional graph convolutional networks (GCN) [5]. This approach enables the model to adapt to dynamic changes caused by traffic rules and events, resulting in a more effective capture of spatial correlations in traffic data.

(2) Long-term dependency handling. To improve the accuracy of long-term traffic trend prediction, we combine gated recurrent units (GRU) and self-attention mechanisms. This enhances the model's ability to explore complex dependencies within the time series and more effectively capture long-term dependencies, which are critical for accurate prediction of future traffic flows.

(3) Computational efficiency and performance. While incorporating advanced attention mechanisms and graph attention mechanisms, we prioritize computational efficiency. We simplify the model structure and optimize the algorithm to significantly reduce the computational resource requirements while still maintaining high-performance predictions. This makes the model more practical, feasible, and scalable for real-world applications.

(4) Empirical study. We conduct extensive experimental validation on public datasets, and the results demonstrate that our proposed model outperforms existing methods on both short-term and long-term traffic prediction tasks. Moreover, the model exhibits excellent generalization ability and robustness, further confirming the effectiveness and reliability of our approach.

The rest of the paper is organized as follows. Section 2 presents the research progress and related work. Section 3 provides definitions of the basic concepts. Section 4 presents the specific structure and implementation of the AMSTGCN model. Section 5 discusses the experimental validation and analysis of the results. Section 6 provides a summary and concluding remarks.

2 Related Work

With advances in science and technology, methods for traffic prediction have evolved from traditional statistical models to modern machine learning and deep learning models. In the early stages, statistical methods such as the Historical Average (HA) model and the ARIMA model were predominant. These models are typically applicable to linear time series data, but their predictive power is limited for complex, high-dimensional, externally influenced traffic data. In addition, the parameters of these methods often rely on expert knowledge for manual configuration, rather than being obtained through data-driven self-learning training.

In the wave of artificial intelligence, various machine learning methods, including K-Nearest Neighbors (K-NN) [6], Support Vector Regression (SVR) [7], Random Forest [8], and Bayesian Neural Networks [9] have been widely adopted in the field of traffic prediction. These techniques enhance prediction accuracy by exploring data features in-depth and typically optimize performance through the combination of different algorithms. However, they typically perform poorly in capturing long-term dependencies in traffic data, which is crucial for understanding traffic patterns and trends [10].

The rise of deep learning has brought revolutionary advances in traffic prediction. CNN benefiting from their remarkable achievements in image processing, has been applied in time series analysis. Researchers transform the spatial-temporal characteristics of traffic flow data into two-dimensional matrices (similar to images) and use CNN to extract features from these "spatial-temporal images" to achieve accurate prediction of speeds on extensive road networks [11]. However, CNN lacks mechanisms to handle long-term temporal dependencies, which may limit its effectiveness in predicting long-term trends.

To overcome the issue of long-term dependencies, Recurrent Neural Networks (RNN) have been introduced in traffic prediction, as they possess the ability to handle sequential data with memory [12]. However, inherent problems of RNN, such as vanishing gradients and exploding gradients, limit their performance in modelling long sequences. Therefore, improved versions of RNN, such as GRU and Long Short-Term Memory (LSTM) [13], have been more widely used in traffic prediction due to their structural advantages and stronger performance in addressing these challenges.

Despite extensive exploration of traffic prediction methods, traditional techniques have focused on temporal relationships, overlooking the significant impact of spatial dynamics on traffic patterns. To rectify this oversight, GCNs have been increasingly utilized for modelling and extracting spatial interrelations. GCNs have inherent strengths in representing non-Euclidean irregular graphs and capturing spatial correlations by aggregating information from nodes and their surroundings, rendering them particularly suitable for traffic network analysis. As a result, a surge of GCN-based traffic flow prediction models has surfaced, including T-GCN [14], STGCN [15], and STSGCN [16], among others. To further refine the capture of intricate spatial-temporal dependencies, models that incorporate attention mechanisms, such as GAT [17], have been developed and integrated into frameworks like ASTGCN [18], GAGCN [19], STN-GCN [20], along with other approaches [21, 22]. These sophisticated models combine Transformer, GCN, GRU, and additional architectures to achieve exceptional prediction capabilities. Nonetheless, as the complexity of these models escalates with the number of modules and depth, so does the computational intensity and the demand for resources.

Our study proposes an innovative model for traffic prediction that aims to combine the attention mechanism with GCN and GRU to intensively explore the relationships between data in both temporal and spatial dimensions. Our goal is to achieve prediction performance comparable to or even better than that of a complex model with a relatively simple model structure. The proposed model demonstrates significant advantages in three key aspects.

Firstly, in terms of spatial-temporal relationship modelling, we employ an improved Graph Attention Network (GAT_v2) instead of the traditional GCN approach based on a predefined adjacency matrix. GAT_v2 can adaptively learn dynamic relationships between nodes, which is particularly suitable for traffic prediction as it can accommodate spatial relation changes in the traffic network caused by regular variations or unexpected events. Secondly, to address the issue of long-term dependencies in time-series data, our model combines GRU with self-attention mechanisms. Compared to existing methods based on Long Short-Term Memory (LSTM), our combined approach is not only more concise but also performs equally well in handling lengthy sequences, providing an effective alternative. Finally, we carefully consider the computational efficiency during the design of our method, which is particularly important for resource-constrained scenarios. By optimizing the computational workflow, our model reduces the need for computational resources while maintaining high prediction performance. Compared to complex deep learning models, our approach demonstrates better practicality and scalability.

3 Problem Definition

Traffic flow refers to the movement and flow of vehicles, pedestrians, or goods in a transportation system. Traffic flow is characterized by its volume, speed, and density. These features reflect the congestion level, mobility, and efficiency of the transportation system. By managing and optimizing traffic flow, we can improve the operational efficiency and travel experience of the transportation system. Attention mechanisms have the potential to extract spatial-temporal relationships, so our study focuses on verifying their role in traffic flow prediction and evaluating the performance of designed models. To eliminate noise and perturbations caused by multiple input features, we specifically choose to predict traffic speeds and conduct experiments to visually illustrate the prediction results. Of course, our model is also applicable to predicting traffic volume and traffic density.

To capture the spatial correlation of traffic speed data, we define the road network as a graph structure. $\mathrm{G}=(\mathbf{V},\mathbf{E},\mathbf{A})$, Where G represents the road network graph, E represents the edge, $|\mathbf{V}|=N$ is the number of road nodes, and $\mathbf{A}\in R^{N\times N}$ is the adjacency matrix reflecting the connectivity relationship between nodes. The elements may be denoted by 0, 1, and may also be measured by distances. $\mathbf{X}$ represents the input characteristic matrix, X^{T_P} represents the feature at the P-th time step. $x_N^{T_P}$ denotes the input feature of the N-th node at the P-th time step. Here the features can be multi-dimensional, that is, speed, flow, density, etc. $\mathbf{X}$ can be expressed as Eq. (1):

$$\mathbf{X}=(X^1,X^2,...,X^{T_P})=\begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^{T_P} \\ x_2^1 & x_2^2 & \cdots & x_2^{T_P} \\ \vdots & \vdots & \ddots & \vdots \\ x_N^1 & x_N^2 & \cdots & x_N^{T_P} \end{bmatrix}. \quad \mathbf{(1)}$$

The traffic speed prediction problem becomes learning a function that can map the past P historical graphs to the future Q graphs given the known graph structure, which can be expressed as Eq. (2):

$$[X^{(t-T_p+1)},\cdots,X^t;\mathbf{G}] \xrightarrow{f(\cdot)} [X^{(t+1)},\cdots,X^{(t+T_Q)}]. \quad \mathbf{(2)}$$

4 Entire Structure

To effectively capture the spatial-temporal correlations in traffic data, this study proposes a spatial-temporal relationship extraction model called AMSTGCN. As shown in Fig. 3, this model consists of four core components: input module, spatial relation extraction module, temporal relation extraction module, and output module. In the input module, we perform a series of preprocessing operations on the raw traffic data to adapt to the requirements of the subsequent prediction task. These preprocessing steps include padding the missing data, removing outliers, normalizing the data, and partitioning the dataset to ensure data quality and efficient model training. The specific preprocessing method is detailed in Section 4.1 of the paper. For spatial relationship extraction, we utilize an upgraded version of GAT called GAT_v2. Compared to the original static attention mechanism in GAT, GAT_v2 can dynamically capture attention relationships, which is a significant advantage for complex and dynamic traffic road operation environments. The implementation details of this module are discussed further in Section 4.2. The temporal correlation extraction module uses gated units to capture the dependencies in the time series and combines the attention mechanism to compute the attention coefficients, thus revealing the temporal correlations between the data accurately. The specific implementation of this part is explained in detail in Section 4.3. Finally, in the output layer, we design a fully connected layer to generate multi-step prediction results. Through an organic combination of these four modules, the AMSTGCN model achieves high-precision traffic flow prediction.

4.1 Input Layer

In a traffic scenario, traffic features can include traffic speed, traffic flow, and lane occupancy. Any of these features can be chosen for traffic flow prediction. Typically, in short-term traffic flow prediction, equidistant sampling is done at intervals of 5 minutes, 10 minutes, or 15 minutes. However, traditional traffic data collectors are prone to faults, such as communication issues, power supply problems, and road maintenance, which can result in missing or abnormal data. To ensure the accuracy of subsequent predictions, the collected data needs to be reprocessed. For outliers and missing data, padding is done by computing the historical average. To make the input a feature representation that can participate in the computation of the GAT network, the form of the input data has been adjusted as Eq. (3):

$$\mathbf{X} = [X_1, X_2, \cdots X_N] = \begin{bmatrix} x_1^{T_1} & x_2^{T_1} & \cdots & x_N^{T_1} \\ x_1^{T_2} & x_2^{T_2} & \cdots & x_N^{T_1} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{T_P} & x_2^{T_P} & \cdots & x_N^{T_P} \end{bmatrix} . \tag{3}$$

X_N represents the N-th node feature and $x_N^{T_P}$ represents the feature value of the N-th node at the P-th time step. $\mathbf{X} \in \mathbf{R}^{\mathbf{N}}$ is the node input feature that satisfies the GAT network operation and will be fed into the subsequent spatial relation extraction layer to obtain the spatial correlation and complete the node state update.

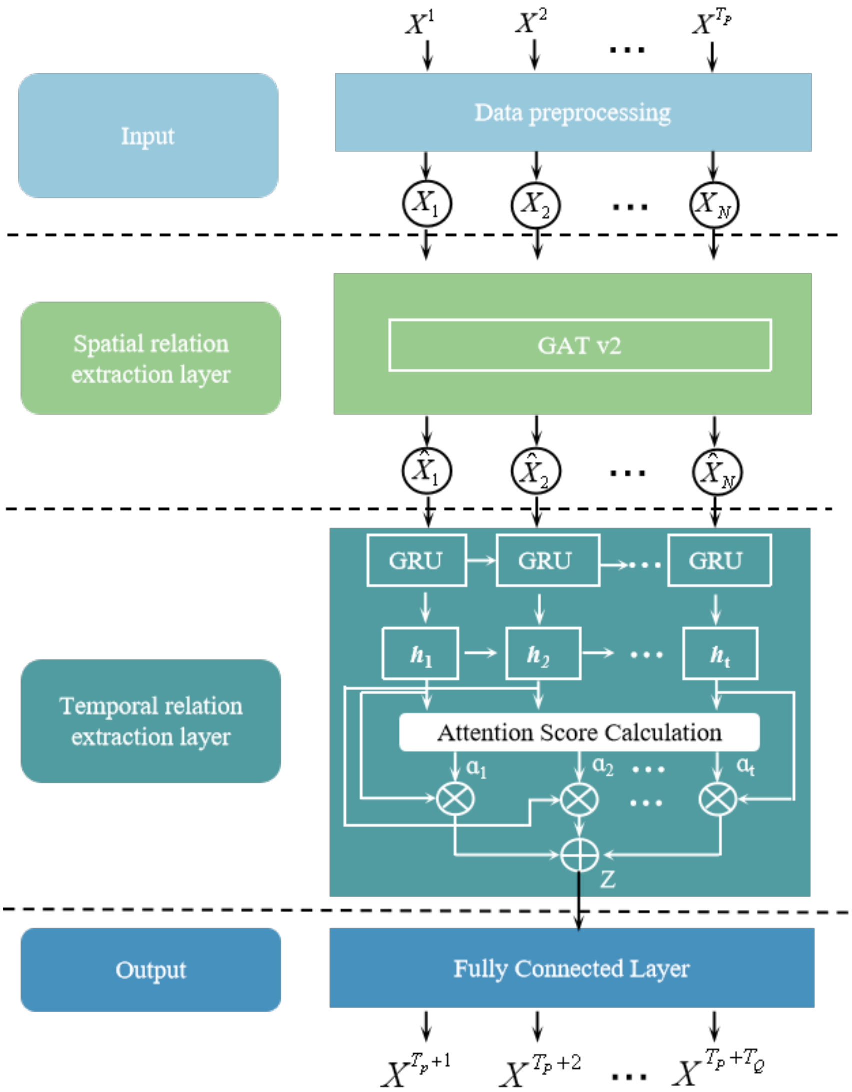


Fig. 3. Block diagram of the overall structure of AMSTGCN

4.2 Spatial Relation Extraction Layer

In the context of transportation, spatial relationships between road nodes are not only characterized by fixed spatial locations but also exhibit dynamic dependencies that shift over time. Therefore, it is crucial to obtain dynamic and adaptive relations that can account for scenario variations. GAT introduces an attenuation mechanism that allows each node to focus on its neighbours to varying degrees depending on their importance. This allows GAT to capture interactions between nodes more accurately, instead of merely averaging or weighting neighboring nodes as is done in GCN. In addition, GAT supports multi-head attention, meaning that multiple attention heads can be used simultaneously to learn the relationships between nodes. This approach enhances the expressive power of the model and enables a better capture of complex relations within the graph structure. However, reference [4] demonstrates that for a fixed set of GAT keys, the resulting attention coefficient remains relatively invariant if attention is computed using different queries on this set of keys. In other words, the ordering of attention coefficients is the same for all nodes in the graph and independent of the query node. This implies that the attention computation function is static and does not change with different queries. This is a problem with the GAT model, which significantly reduces the expressive power of GAT. To obtain a dynamic attention mechanism, a modified GAT model GAT_v2 is used in this paper. The improvement of GAT_v2 over GAT is shown in Eq. (4):

$$\begin{cases} \text{GAT} \quad \rightarrow e_{ij} = \text{LeakyRelu}\left(\alpha^{\mathrm{T}} \cdot \left[Wx_i \,\|\, Wx_j\right]\right) \\ \text{GATv2} \rightarrow e_{ij} = \alpha^{\mathrm{T}}\ \text{LeakyRelu}\left(W \cdot [x_i \,\|\, x_j]\right) \end{cases} \tag{4}$$

Observing Eq. (4), we find that GAT_v2 only modifies the order of the internal operations of GAT to play the role of repairing the attention function. Readers interested in the specific proof procedure for the GAT_v2 model can refer to reference [4], which we directly cite and apply in this paper. Fig. 4 shows the complete computational process of updating node features using GAT_v2 as an example of node *i*.

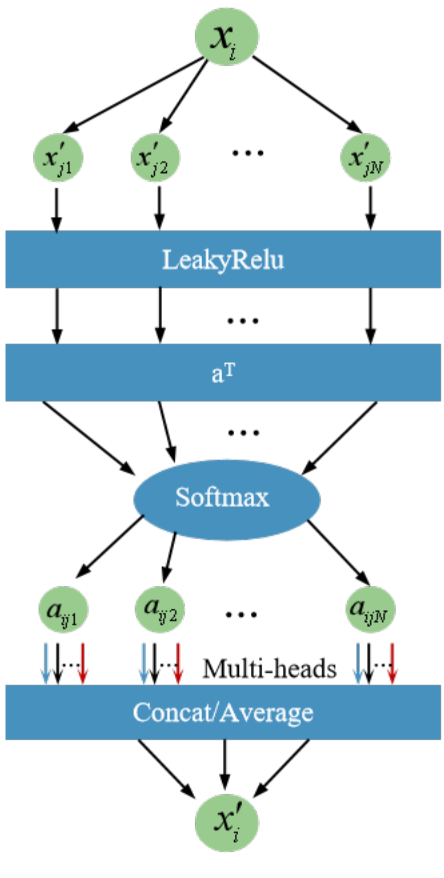


Fig. 4. Flowchart of GAT_v2 attention mechanism computation

GAT_v2 can be implemented by Eq. (5) ~ Eq. (7).
(1) Calculate the attention coefficient.

$$e_{ij} = \alpha^{\mathrm{T}} \operatorname{LeakyRelu}\left(W \cdot [x_i \,||\, x_j]\right). \tag{5}$$

e_{ij} represent the attention value of node *i* relative to node *j*. α^{T} and W is shared learning parameters. || denotes the vector concatenation. This expression means that when computing the attention coefficients, the linear transformation is applied after concatenation, the nonlinear computation is done by the activation function, and finally, the transformation is applied. In this way, it can be conditioned on the query node and finally implement the computation of dynamic attention.
(2) The attention coefficients are normalized by softmax.

$$\alpha_{ij} = \operatorname{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}. \tag{6}$$

(3) Node character updates.

$$x_i' = \sigma\left(\sum_{j \in N_i} \alpha_{ij} \cdot W x_j\right). \tag{7}$$

x_i' represents the current feature of node i after the fusion of neighbourhood information. σ is the activation function.
To enhance the ability to obtain spatial correlations, a multi-head attention mechanism is used. Since the final output is not the final result of our prediction, which is in the middle layer of the model, we employ a concatenation method such as Eq. (8). Of course, the sum-and-average approach can also be adopted depending on the different tasks, as shown in Eq. (9).

$$x_i' = \mathop{\Big\|}_{K=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^{k} \cdot W^{k} x_j\right). \tag{8}$$

$$x_i' = \sigma\left(\frac{1}{K} \sum_{K=1}^{K} \sum_{j \in N_i} \alpha_{ij}^{k} \cdot W^{k} x_j\right). \tag{9}$$

4.3 Temporal Relation Extraction Layer

In the time-related feature extraction module, we use a combination of GRU and self-attention mechanisms. Compared to RNN and LSTM, GRU has a simpler structure and a memory mechanism, making it suitable for long and short-term time series prediction. The self-attention mechanism can also extract correlations between each time step.
The structure of the GRU is shown in Fig. 5. In GRU, the update gate and reset gate are two essential gating mechanisms to control the flow and update of information. The role of the update is to determine the weight of the hidden state of the input at the current moment and the previous moment, and at the previous moment to decide whether the hidden state of the input needs to be updated. The role of the reset gate is to decide how the input information at the current moment interacts with the hidden state at the previous moment.
GRU is calculated as shown in Eq. (10):

$$\begin{cases} z_t = \sigma\left(W_Z \cdot \left[h_{t\text{-}1}, x_t\right] + b_z\right). \\ r_t = \sigma\left(W_r \cdot \left[h_{t\text{-}1}, x_t\right] + b_r\right). \\ \tilde{h}_t = \tanh\left(W_c \cdot \left[r_t \odot h_{t\text{-}1}, x_t\right] + b_c\right). \\ h_t = (1 - z_t) \odot h_{t\text{-}1} + z_t \odot \tilde{h}_t. \end{cases} \quad \textbf{(10)}$$

Where z_t is the update gate, r_t is the reset gate, $\tilde{h}_t$ is the candidate hidden state, $h_{t\text{-}1}$ is the hidden state at the previous time step, h_t is the hidden state at the last time step, $\odot$ is the Hadamard product, which stands for element-wise multiplication. σ and $\tanh$ are the activation function. W_z, W_c, and W_r are weight parameters. b_z, b_r, and b_c are the bias parameter.

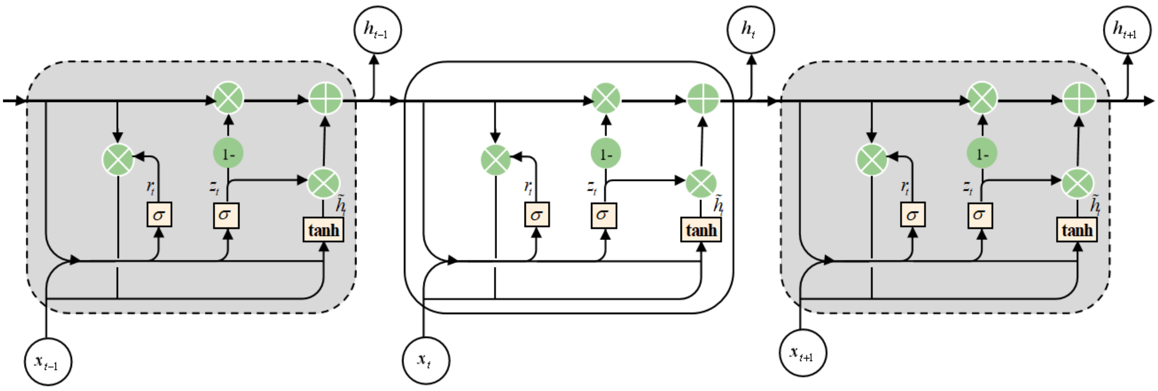


Fig. 5. The basic structure diagram of GRU

After the GRU calculation, we can obtain the hidden states at all-time steps. To further obtain the long-range dependence, we perform an attention calculation. The attention score calculation process is shown in Fig. 6, and the calculation steps are shown in Eq. (11):

$$\begin{cases} e_t = \tanh(W_e h_t + b_e). \\ \alpha_t = \mathrm{softmax}(e_t). \\ z = \sum_{i=t-P+1}^{t} \alpha_i h_i. \end{cases} \quad \textbf{(11)}$$

Where h_t is the hidden state output of the GRU, W_e and b_e are the learnable weight and bias parameters, respectively.

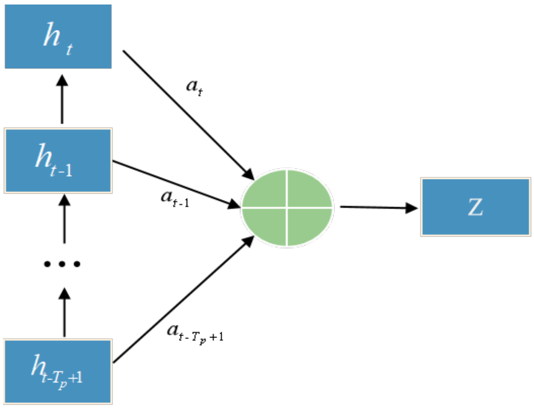


Fig. 6. Schematic of the attention mechanism followed by GRU

4.4 Output Layer

Our goal is to predict the traffic flow for Q steps into the future. Therefore, a fully connected layer is used in the output layer to complete the dimensional transformation.

$$Y = \mathrm{Relu}(W_o Z + b_o) \tag{12}$$

The fully connected input is the attention value Z obtained by the time extraction layer, $W_O \in R^{N\times Q}$ is the learnable parameter and the output of the output layer is the traffic feature of each node in the future Q time steps.

4.5 Loss Function

During the model training process, the primary objective is to minimize the discrepancy between the observed traffic speed and the predicted values generated by the model. We denote the true traffic speed Y and the predicted traffic speed by $\hat{Y}$. The loss function as Eq. (13):

$$Loss = \left\| Y - \hat{Y} \right\| + \lambda L_{reg} \tag{13}$$

The first term in Eq. (13) is used to calculate the difference between the actual traffic velocity and the expected velocity. The next component is the L2 regularization component, which is used to control the complexity of the model and λ is a hyperparameter.

5 Experiments

5.1 Datasets and Experimental Settings

To evaluate the performance of our model, we conduct experiments using the publicly available Loop Seattle dataset. This dataset was collected by the Seattle Department of Transportation and consists of traffic speed data from 323 sensor stations located on highways in the Seattle area (I-5, I-405, I-90, and SR-520). The data spans the entire year 2015 and is collected at a resolution of 5 minutes [23].

In the experiments, we split the datasets using a ratio of 0.7:0.1:0.2, which means the Train dataset: Valid dataset: Test dataset = 0.7: 0.1: 0.2. The data is normalized after partitioning and we use z-score normalization method for normalization: $\overline{x} = (x - \mu)/\sigma$, x_{max} is the maximum and x_{min} is the minimum of the sample data.

The model is developed using the PyTorch 1.9.0 deep learning framework. The specific configuration information is as follows: CPU: Intel(R) Core(TM) i7-7800X，24GB Graphics Card: GeForce RTX 3090, CUDA version: 11.3.

5.2 Evaluation Metrics

To evaluate the performance of the AMSTGCN model, we use two evaluation metrics, namely Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y - \hat{Y}\right| . \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y - \hat{Y})^2} . \tag{15}$$

Y represents the true traffic speed, $\hat{Y}$ represents the predicted traffic speed. The smaller MAE and RMSE demonstrate the better prediction performance of the model.

5.3 Experiment and Result Analysis

To evaluate the performance of the model, we performed a series of experiments, including comparing the predictive power of the model to the baseline model, analyzing the effect of different components on the model performance, and measuring the computational time cost.

(1) Comparison experiments with baseline models.

Our task is to predict future velocities at the 3rd, 9th, and 12th time points using the known velocity values at the past 12 sampling points. Given that the raw data is sampled at 5-minute intervals, this amounts to predicting the next 15, 45, and 60-minute velocities based on historical velocity data from the past hour. To evaluate the model performance, we compare AMSTGCN with five baseline models. The comparison of the prediction performance of the AMSTGCN model on the LOOP_SEATTLE dataset with the five baseline methods is presented in Table 1.

Table 1. The prediction performance of the AMSTGCN model and other baseline methods on the LOOP_SEATTLE dataset

Methods	LOOP_SEATTLE					
	15min		45min		60min	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
HA	5.32	8.96	5.32	8.96	5.32	8.96
FNN	3.17	5.99	4.45	8.13	4.99	9.05
GRU	4.27	7.67	4.40	7.93	4.52	8.14
T-GCN	3.65	5.95	4.86	7.84	5.32	8.64
DCRNN	<u>2.94</u>	5.96	4.07	7.33	4.40	8.15
AMSTGCN	3.73	<u>5.94</u>	<u>4.06</u>	<u>6.62</u>	<u>4.21</u>	<u>6.91</u>

HA: History Average Model. It predicts future observations based on the average value of past observations over a certain period.

FNN: Fully connected Neural Network. We constructed the simplest three-layer fully connected neural network to verify the prediction performance of the simple model.

GRU: Gated Recurrent Unit employs gate mechanisms to regulate information flow, mitigating gradient vanishing and exploding issues prevalent in traditional RNNs.

T-GCN [14]: Temporal Graph Convolutional Network exploits graph convolutions to discern node interactions and GRUs to apprehend temporal dynamics.

DCRNN [24]: Diffusion Convolutional Recurrent Neural Network synergizes diffusion convolution with recurrent networks to model the spatial-temporal dynamics inherent to traffic networks.

Based on the experimental data, the AMSTGCN model shows significant advantages in spatial-temporal prediction tasks. By combining the spatial correlation acquisition capability of GAT_v2 with the temporal feature extraction advantage of the attention mechanism, this model significantly improves the predictive performance across different time scales. Specifically, AMSTGCN does not outperform DCRNN, FNN, and T-GCN when predicting 15 minutes, but its MAE of 3.73 and RMSE of 5.94 are still quite impressive. This indicates that AMSTGCN can provide competitive results even within a relatively short prediction window.

The advantage of AMSTGCN becomes apparent as the prediction horizon extends to 45 minutes. MAE was reduced to 4.06 and RMSE was further reduced to 6.62, outperforming all compared models. This suggests that AMSTGCN has a stronger ability to capture and exploit long-term dependencies in the data.

The advantage of AMSTGCN becomes even more prominent at the 60-minute prediction point. It achieves an MAE of 4.21 and RMSE of 6.91, again showing the lowest error rate among all models. This significant performance improvement is attributed to the deep spatial relationship mining capability of GAT_v2 and the flexibility provided by the attention mechanism in handling temporal information.

In summary, the AMSTGCN model not only maintains good performance in short-term prediction but also demonstrates excellent capabilities in long-term prediction. This is driven by the carefully designed model structure, in particular the efficient integration of GAT_v2 and attention mechanisms, which enables AMSTGCN to accurately capture the crucial spatial-temporal dynamics in complex data. As a result, it achieves higher accuracy and reliability in future predictions. This has practical implications and applications in domains that require accurate spatial-temporal predictions, such as traffic management, weather forecasting, and urban planning.

The predictions for node 10 and node 320 in the LOOP_SETTLE dataset are displayed in Fig. 7 and Fig. 8, which help to make the prediction results more understandable.

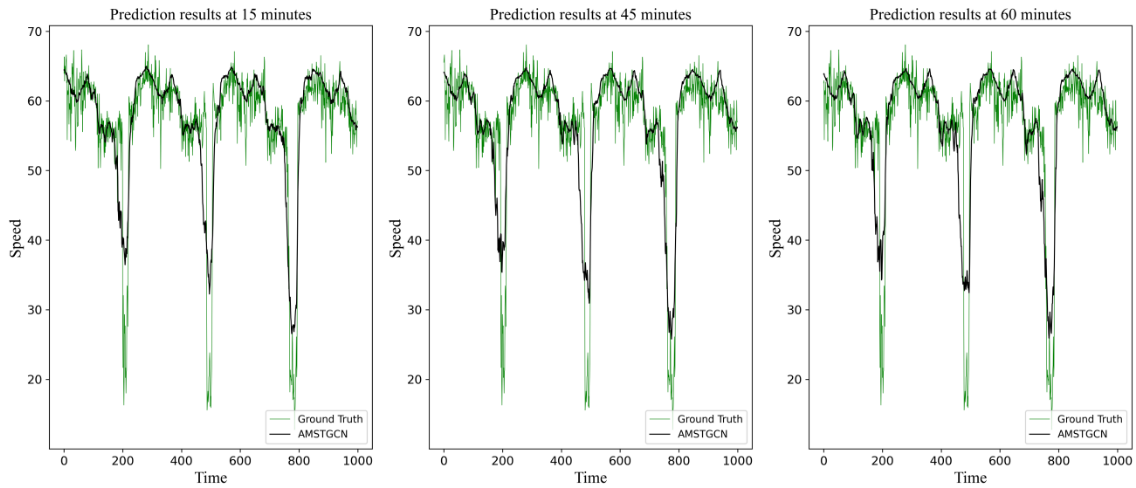


Fig. 7. Visualization of prediction results for node 10 in the LOOP_SEATTLE dataset

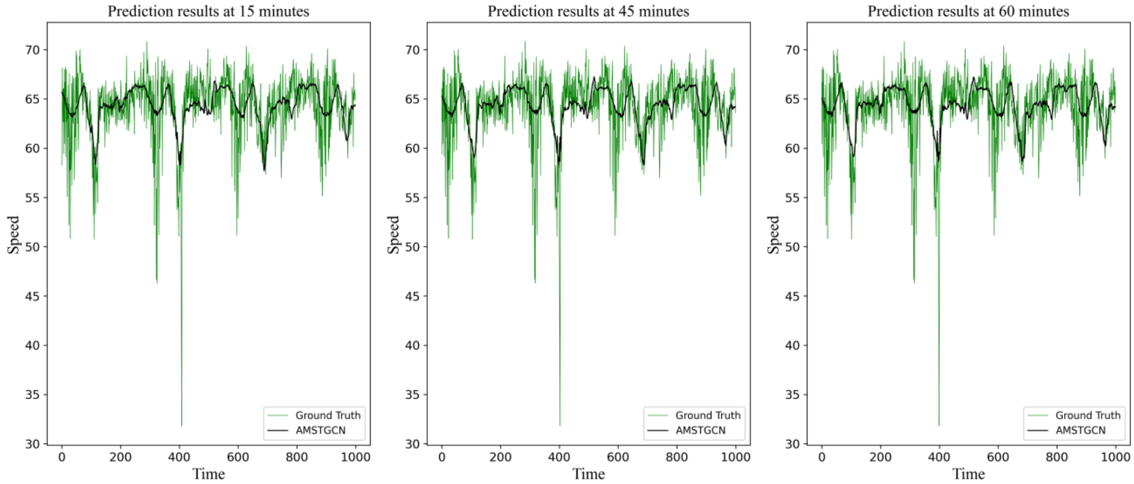


Fig. 8. Visualization of prediction results for node 320 in the LOOP_SEATTLE dataset

(2) Performance testing of models with different components.

To deeply investigate the specific role of GAT_v2 and the attention mechanism in spatial-temporal prediction models, we conduct a series of comparative experiments. These experiments aim to assess the contribution of each component in capturing spatial-temporal correlations through different combinations of models. Specifically, we integrated and combined baseline models such as GAT, GAT_v2, GRU, and Attention Mechanisms to construct various hybrid models. These hybrid models are designed to reveal the unique value and synergistic impact of each module in integrating spatial-temporal information. For ease of comparison and understanding, we present the structure of different model combinations in Table 2. In addition, to simplify the exposition and aid the reader's understanding, we refer to the AMSTGCN model as G2GA.

Table 2. Models and naming of different combinations

Model name	GG	GGA	G2G	G2GA (AMSTGCN)
Combination	GAT+GRU	GAT+GRU+Attention	GAT_v2+GRU	GAT_v2+GRU+Attention

With these comprehensive tests, we aim to demonstrate the advantage of GAT_v2 in spatial relation mining and the effectiveness of the attention mechanism in extracting temporal sequence features. We also compare the MAE and RMSE values predicted for different time points. The performance tests for the models with different components are shown in Table 3.

We further examine the performance of different models at prediction intervals of 15, 30, 45, and 60 minutes. At the 15-minute point, the G2G model performs slightly better than the others, but the AMSTGCN model is also very close in performance. However, as the prediction duration increased, especially in the 60-minute forecast task, we observed that the AMSTGCN model has lower MAE and RMSE values compared to the other three models, with respective values of 4.21 and 6.91. This indicates that the AMSTGCN model is more effective in capturing and exploiting complex patterns within spatial-temporal data, especially for long-term prediction. This can be attributed to the integration of GAT_v2 and attention mechanisms in the AMSTGCN model, which are better equipped to capture spatial relationships and temporal sequential features in spatial-temporal data. GAT_v2 has the advantage of mining spatial relationships to effectively capture correlations between geographic locations, while the attention mechanism can weight information across different time steps in the temporal dimension to extract salient features from the time series. With this combination, the AMSTGCN model can predict future spatial-temporal changes with greater accuracy. This has significant practical implications for applications in various fields such as traffic flow prediction, weather prediction, and human motion prediction.

Overall, through a comprehensive comparison and analysis of different models, we have validated the specific role of GAT_v2 and the attention mechanism in spatial-temporal prediction models. As a hybrid model integrating these two components, the AMSTGCN model demonstrates superior performance in the long-term spatial-temporal prediction task. These findings provide valuable references and guidance for further research and applications of spatial-temporal prediction models. The visual comparison results for MAE and RMSE are shown in Fig. 9.

Table 3. Performance testing of models with different components

Methods	LOOP_SEATTLE							
	15min		30min		45min		60min	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GG	3.76	5.92	3.96	6.37	4.12	6.72	4.31	7.10
G2G	3.67	5.83	3.90	6.36	4.09	6.77	4.27	7.10
GGA	3.72	5.95	3.89	6.33	4.07	6.68	4.24	6.98
G2GA (AMSTGCN)	3.73	5.94	3.90	6.29	4.06	6.62	4.21	6.91

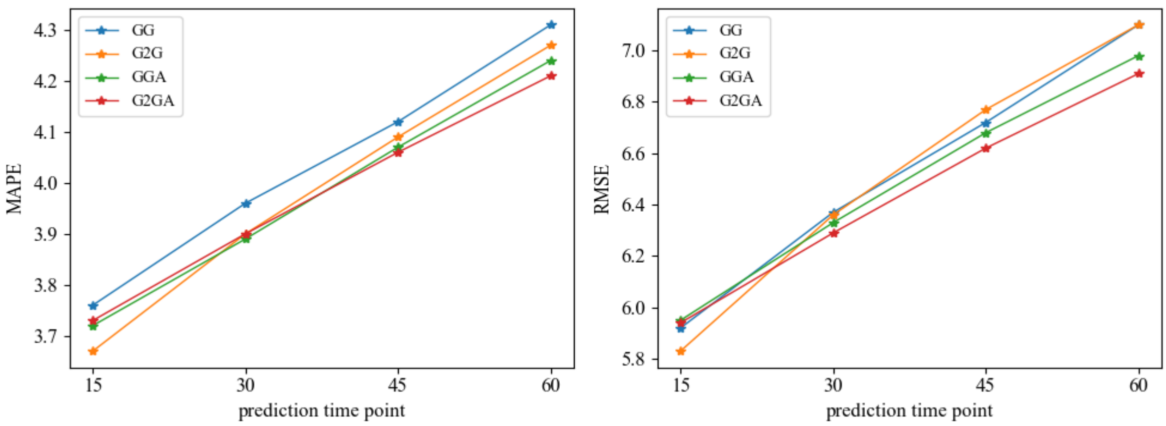


Fig. 9. Changes in performance metrics of different models for prediction tasks of different time lengths

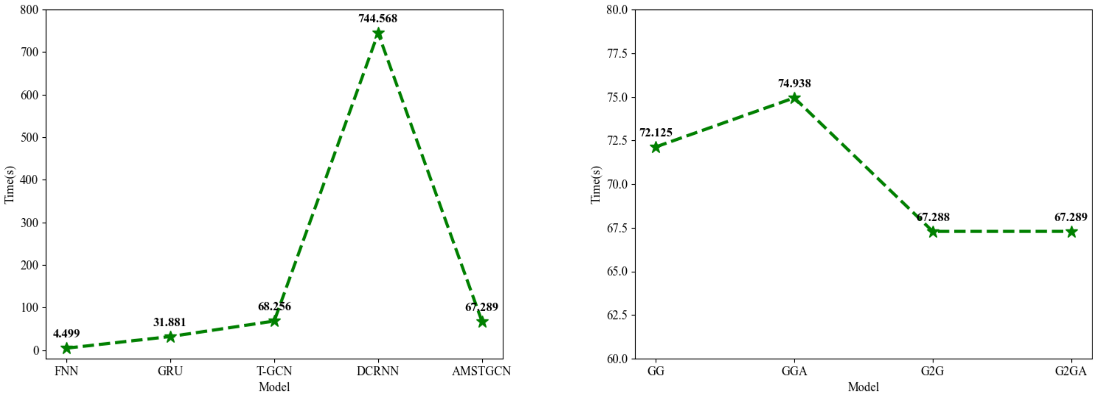
(3) Model training time measurements.

To demonstrate the computational performance advantage of AMSTGCN, we compare its training time with different models. Fig. 10(a) illustrates the comparison of AMSTGCN's training time with four baseline models. We can see that AMSTGCN has a relatively short training time of 67.289 seconds. By comparing the results, we can observe that while the training time of AMSTGCN is longer than FNN and GRU, it is significantly shorter than DCRNN, which requires the most training time. In addition, AMSTGCN also has a slightly shorter training time compared to T-GCN. This indicates that even though AMSTGCN is more complex than some simple models like FNN, it is more efficient in handling complex graph data.

Fig. 10(b) illustrates the training time with different components. When comparing the number of training epochs of different models, we find that the G2G model has a significant computational efficiency advantage over the GG model. Specifically, the training time of the G2G model is 67.288s, while the training time of the GG model is 72.125s. This indicates that the GAT_v2 version is more efficient than the original GAT version, saving approximately 4.837s of training time without introducing the Attention mechanism. When we introduce the attention mechanism into both GG and G2G models, we observe an increase in training time. This is because the attention mechanism adds complexity and computational overhead to the model. However, even after incorporating the Attention mechanism, the training time of the AMSTGN model remains highly close to that of the G2G model with only a 0.001s increase. This minor increase is negligible and suggests that the AMSTGN model maintains strong computational efficiency while improving model performance with the Attention mechanism. In contrast, the training time of the GGA model is significantly increased to 74.938s, which is an increase of 2.813s compared to the GG model. This increase in time may be attributed to the lower computational efficiency of the

GAT version when dealing with the Attention mechanism.

Therefore, we can conclude that the GAT_v2 version provides higher computational efficiency under the same conditions, while the AMSTGN model maintains acceptable computational efficiency while incorporating the Attention mechanism.



(a) Training time with baseline models

(b) Training time with different components

Fig. 10. Changes in performance metrics of different models for prediction tasks of different time lengths

We can conclude that the advantage of AMSTGN lies in its ability to capture temporal dependencies and graph structural features, which allows it to maintain relatively high accuracy while effectively controlling the computational cost. Therefore, AMSTGCN should be a relatively good choice if the application scenario requires taking into account the dynamic nature of the graph and the complex interactions between nodes.

6 Conclusions

This study presents a novel model that integrates GAT_v2 and GRU to address the core issues in traffic prediction. This model goes beyond the limitations of traditional GCN in modelling spatial-temporal relationships. By adaptively learning the dynamic relationships between nodes, it effectively handles spatial relation changes caused by regular variations or unexpected events in the traffic network. Moreover, the model combines the self-attention mechanism and GRU to elegantly address the problem of long-term dependencies in time series data. Moreover, the model is designed with a focus on computational efficiency, optimizing the computational process to adapt to resource-constrained real-world applications while maintaining strong predictive performance. Experimental results on public datasets validate the superior performance of the proposed model compared to existing methods on short-term and long-term traffic prediction tasks and also demonstrate its excellent generalization ability and robustness.

Future work can further expand and deepen the achievements of this study in several directions. First, explore the integration of this model with different types of spatial-temporal data modelling approaches, such as introducing multi-scale analysis or considering more complex spatial-temporal relationships. Second, given the diversity of real-world traffic scenarios, the adaptability and robustness of models remain crucial research topics that can be tested and improved by introducing more diverse datasets and scenarios. Finally, as computational resources continue to evolve, exploring how to leverage parallel computing and distributed systems to tackle larger-scale traffic prediction problems is also an essential direction for future research.

7 Acknowledgement

This research was funded by the Natural Science Foundation of Gansu Province under Grant (22JR5RA186), the

Fundamental Research Funds for the Central Universities under Grant (31920230139), and the National Natural Science Foundation of China (62366048).

References

[1] D.A. Tedjopurnomo, Z.F. Bao, B.H. Zheng, F. Choudhury, A.K. Qin, A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges, IEEE Transactions on Knowledge and Data Engineering 34(4) (2022) 1544-1561.

[2] J. Ye, J. Zhao, K. Ye, C. Xu, How to Build a Graph-Based Deep Learning Architecture in Traffic Domain: A Survey, IEEE transactions on intelligent transportation systems 23(5)(2022) 3904-3924.

[3] Y. Ma, S. Cheng , Y. Ma , Y. Ma, Convolutional Neural Networks and their applications in Intelligent Transportation Systems: A Survey, Journal of Transportation Engineering 21(4)(2021) 24.

[4] S. Brody, U. Alon, E. Yahav, How Attentive are Graph Attention Networks?, in: Proc. 2022 International Conference on Learning Representations(ICLR), 2022.

[5] T.N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: Proc. 2017 International Conference on Learning Representations(ICLR), 2017.

[6] T.H.H. Aldhyani, M.R. Joshi, Enhancement of Single Moving Average Time Series Model Using Rough k-Means for Prediction of Network Traffic, International Journal of Engineering Research and Applications 7(3)(2017) 45-51.

[7] Z.E. Abou Elassad, H. Mousannif, H. Al Moatassime, A proactive decision support system for predicting traffic crash events: A critical analysis of imbalanced class distribution, Knowledge-Based Systems 205(2020) 14.

[8] Y. Wen, R. Wu, Z. Zhou, S. Zhang, S. Yang, T.J. Wallington, W. Shen, Q. Tan, Y. Deng, Y. Wu, A data-driven method of traffic emissions mapping with land use random forest models, Applied Energy 305(2022) 117916.

[9] D. Wang, C. Wang, J. Xiao, Z. Xiao, W. Chen, V. Havyarimana, Bayesian optimization of support vector machine for regression prediction of short-term traffic flow, Intelligent data analysis 23(2)(2019) 481-497.

[10] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, B. Yin, Deep Learning on Traffic Prediction: Methods, Analysis, and Future Directions, IEEE Transactions on Intelligent Transportation Systems 23(6)(2022) 4927-4943.

[11] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction, Sensors 17(4)(2017) 818.

[12] R. Bikmukhamedov, A. Nadeev, G. Maione, D. Striccoli, Comparison of HMM and RNN models for network traffic modeling, Internet Technology Letters 3(2)(2020) e147.

[13] X. Zhang, Q. Zhang, Short-Term Traffic Flow Prediction Based on LSTM-XGBoost Combination Model, Computer Modeling in Engineering & Sciences 125(1)(2020) 95-109.

[14] L. Zhao, Y. Song, C. Zhang, Y. Liu, H. Li, T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction, IEEE Transactions on Intelligent Transportation Systems (99)(2019) 1-11.

[15] B. Yu, H.T. Yin, Z.X. Zhu, Spatial-temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, in: Proc. 27th International Joint Conference on Artificial Intelligence (IJCAI), 2018.

[16] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting, in: Proc. 2020 Association for the Advancement of Artificial Intelligence(AAAI), 2020.

[17] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, in: Proc. 2018 International Conference on Learning Representations(ICLR), 2018.

[18] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting, in: Proc . 2019 Association for the Advancement of Artificial Intelligence(AAAI), 2019.

[19] C. Tang, J. Sun, Y. Sun, Dynamic Spatial-Temporal Graph Attention Graph Convolutional Network for Short-Term Traffic Flow Forecasting, in: Proc. 2020 IEEE International Symposium on Circuits and Systems (ISCAS), 2020.

[20] C. Wang, L. Wang, S. Wei, Y. Sun, B. Liu, L. Yan, STN-GCN: Spatial and Temporal Normalization Graph Convolutional Neural Networks for Traffic Flow Forecasting, Electronics 12(14)(2023) 3158.

[21] Y. Chen, J. Huang, H. Xu, J. Guo, L. Su, Road traffic flow prediction based on dynamic spatiotemporal graph attention network, Scientific Reports 13(1)(2023) 14729.

[22] W. Fang, W. Zhuo, J. Yan, Y. Song, D. Jiang, T. Zhou, Attention meets long short-term memory: A deep learning network for traffic flow forecasting, Physica A: Statistical Mechanics and its Applications 587(2022) 126485.

[23] Z.Y. Cui, K. Henrickson, R.M. Ke, Y.H. Wang, Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting, IEEE Transactions on Intelligent Transportation Systems 21(11)(2020) 4883-4894.

[24] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, in: Proc. 2018 International Conference on Learning Representations (ICLR), 2018.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 August 2023, accepted 2 September 2023, date of publication 6 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312534

RESEARCH ARTICLE

Spatial-Temporal Dynamic Graph Convolutional Neural Network for Traffic Prediction

WENJUAN XIAO [1,2] **AND XIAOMING WANG** [1]

[1]College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
[2]College of Electrical Engineering, Northwest Minzu University, Lanzhou 730030, China

Corresponding author: Wenjuan Xiao (xwj@xbmu.edu.cn)

This work was supported in part by the Natural Science Foundation of Gansu Province under Grant 22JR5RA186, and in part by the Fundamental Research Funds for the Central Universities under Grant 31920230139.

ABSTRACT Due to the complexity and dynamics of transportation systems, traffic prediction has become a challenging task. The accuracy of prediction is influenced by the spatial-temporal correlation within the traffic system. Previous approaches mainly relied on a pre-defined static adjacency matrix combined with graph convolutional neural networks to capture spatial correlation, neglecting the dynamic relationships between nodes over time. In this study, we propose a novel prediction model called the spatial-temporal dynamic graph convolutional neural network (STDGCN). By fusing node embeddings and input features, we obtain a new node representation that incorporates both static and dynamic features. To capture the dynamic relationships, we introduce a similarity calculation to construct a dynamic adjacency matrix. This matrix contains rich spatial relationships that serve as a reference for subsequent prediction tasks. We further employ Graph Convolutional Networks (GCN) and Gated Recurrent Units (GRU) to capture the spatial-temporal correlation. By combining these components, we establish a comprehensive traffic volume prediction model. To evaluate the performance of our proposed method, we conduct experiments on two real datasets. The experimental results demonstrate that our model achieves state-of-the-art performance in accurately predicting traffic volumes.

INDEX TERMS Traffic prediction, spatial-temporal dynamic graph, dynamic adjacency matrix, graph convolutional neural network.

I. INTRODUCTION

Traffic prediction is an essential and integral part of intelligent transportation systems, playing a critical role in optimizing transportation efficiency. Since advanced technologies such as data mining, machine learning, and artificial intelligence can effectively collect and analyze historical traffic data, they are widely used in the field of traffic prediction [1], [2]. There is no doubt that accurate traffic prediction results can provide valuable travel references for traffic participants, which invisibly enhances the overall efficiency of traffic operations and ultimately improves traffic efficiency [3].

Traffic prediction is a multidimensional problem that mainly involves spatial and temporal dimensions, as well as feature dimensions. The spatial-temporal dimension takes into account the variability of traffic volume at different times and locations. To accurately predict traffic conditions, both temporal and spatial dimensions need to be modeled and analyzed. The feature dimension covers various factors that affect traffic conditions, such as road network topology, road conditions, special events, and weather changes. By considering these features in a comprehensive way, we can have a more comprehensive understanding and prediction of traffic conditions. Changes in the spatial-temporal dimension can better reflect the underlying patterns of traffic flow changes, so our study also focuses more on the analysis and modeling of the spatial-temporal dimension.

Traffic data are commonly classified as spatial-temporal data, which implies the presence of both temporal dependencies and location-based dependencies among the data. Additionally, these data exhibit spatial-temporal relationships that are subject to variations over time. In early studies, statistical methods such as the Historical Average(HA)

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

model [4], Vector Auto Regressive model(VAR) [5] and Auto Regressive Integrated Moving Average(ARIMA) model [6] were used for prediction. These methods are based on the assumption of smoothness. They place greater emphasis on temporal correlations in traffic data while disregarding the impact of spatial correlation between transportation systems on on-road operational conditions [7]. Furthermore, the parameters utilized in the modeling are typically determined based on expert knowledge rather than being obtained through iterative optimization. As a result, the performance may be sub optimal when dealing with complex, high-dimensional non-linear traffic data.

Machine learning methods such as K-Nearest Neighbor (KNN) [8], K-Means [9], and others have also been employed for traffic prediction. However, it is worth noting that these methods heavily rely on historical data. In [10], the study of traffic prediction also incorporates the use of Convolutional Neural Network (CNN). However, it is important to note that CNN are primarily utilized for capturing temporal correlations and are limited to processing standard European graph data. In [11], the study of traffic prediction often utilizes Recurrent Neural Network (RNN) in deep learning, such as the Long Short Term Memory (LSTM) [12], GRU [13], and other variants. These methods demonstrate superior performance in capturing temporal correlations and exhibit effectiveness in short-term time series prediction. However, when applied to traffic data that incorporates spatial information, standalone RNN methods frequently overlook the significance of spatial dependencies. The absence of spatial information can lead to a decrease in the accuracy of predictions.

Moreover, in real-world scenarios, the specific location of the data acquisition device also has an impact on the completeness and comprehensiveness of the collected data. To address this, various integrated deep learning models have been applied in traffic data prediction. In [14], a robust deep learning architecture based on stacked Sparse Auto Encoders (SAEs) is proposed to accurately estimate the traffic flow across the entire network using deployed sensor sets. In [15], a deep residual neural network is introduced to reliably forecast the origin-destination (O-D) data of the entire network based on the flow of links. The aforementioned methods are suitable for specific traffic prediction scenarios, but they may not pay sufficient attention to the effects of dynamic spatio-temporal correlations.

The traffic network can be naturally viewed as a graph structure. Given the growing popularity of GCN, they have been extensively utilized in the domain of traffic prediction research [16]. Previous studies on traffic road network structures have predominantly relied on the distance between road nodes as a proxy for representing the spatial correlation between nodes. While this approach is simple and convenient, it fails to capture the dynamic nature of traffic data on road nodes during different time periods or special events, which deviates from real-world traffic scenarios [17]. As shown in Fig. 1, the spatial-temporal correlation of traffic data among residential, school, and commercial areas cannot be solely attributed to spatial proximity. It is also profoundly

FIGURE 1. **Dynamic relationships in traffic scenarios.**

influenced by the functional attributes associated with each area. For instance, on weekdays, there is a significant correlation between schools and residential areas, which naturally diminishes on non-working days. In addition, the occurrence of public gatherings such as sporting events and concerts can result in transient traffic congestion in the area. Such congestion can even extend to remote regions, leading to variations in spatial dependencies between nodes. Consequently, considering static distances alone is insufficient to adequately capture the complex spatial dependencies.

To improve prediction accuracy by capturing more profound and evolving spatial dependencies, we present a novel model called Spatial-Temporal Dynamic Graph Convolutional Neural Network (STDGCN). The main contributions of our study can be summarized as follows:

(1) We propose a joint prediction model based on GCN and GRU, where the generation of the dynamic adjacency matrix is the key link of the model. The input features and node information are encoded and fused, respectively, and then processed by the GRU to obtain the hidden states at different time steps. After fully connected processing of the hidden states at each time step, a cosine similarity calculation is performed to obtain the dynamic adjacency matrix containing rich spatial-temporal information. It more effectively describes the spatial dependencies and dynamics of the road network and improves the predictive performance of the model.

(2) We conduct thorough experiments on three publicly available datasets to demonstrate the efficiency of our proposed model in predicting traffic. Our model outperforms 7 benchmark baselines, exhibits a substantial reduction in prediction error, and achieves the highest level of accuracy in traffic prediction.

The rest of the paper is organized as follows. Section II reviews the development of traffic prediction methods and related work. Section III provides an instantiation analysis of temporal and spatial correlations in traffic data. Section IV presents the specific details of our designed model. Section V evaluates the designed model on two real-world traffic datasets, performs a comparison analysis with baselines, and performs ablation experiments. Finally, Section VI concludes the paper and sets out directions for future research.

II. RELATED WORK

GCN is suitable for non-standard graph structures [18], while road network is a natural graph structure, so it has obvious advantages to use GCN to process traffic data. In the past few years, a growing number of research works have utilized the fusion of GCN and RNN for traffic prediction. The core idea of these models is to use GCN to obtain spatial correlation of traffic data and RNN, GRU, and LSTM to obtain temporal correlation [19]. Models such as temporal convolutional neural networks (TCN) [20] and diffusion convolutional neural networks are employed to capture time-dependent relationships, addressing the challenge of gradient vanishing or exploding in RNN. In the aforementioned approaches, the spatial correlation information derived from GCN is predominantly represented by a predefined adjacency matrix, and the correlations between road nodes are measured by distance weights. In this approach, the transportation network is treated as a static graph that captures only fixed spatial correlations. However, it overlooks any dynamic relationships that may arise from transient congestion, traffic accidents, and other factors [21]. As mentioned in [22], the traffic flow is modeled as a diffusion process on a graph. To capture dynamic spatial correlations, the authors used a combination of diffusion convolutional neural networks and random walk methods. As described in [23], an adaptive adjacency matrix is constructed by utilizing node embedding. This adaptive matrix, in combination with a diffusion convolutional neural network, effectively captures the dynamic correlation between nodes. All the aforementioned models take note of the dynamic and changing character of the spatial-temporal graph. Dynamic graphs have been further investigated in several subsequent models, including ASTGCN [24], STSGCN [25], AGCRN [26], and ESTNet [27]. The ASTGCN and STSGCN models introduced the attention mechanism to enhance their performance. In the AGCRN model, a parameter-adaptive generative approach was employed to generate dynamic graphs. On the other hand, the ESTNet model utilized a multilayer graph structure fusion method to obtain dynamic correlations. The above methods perform well in extracting dynamic spatial-temporal correlations, but most of them build complex models at the expense of efficiency, such as introducing an attention mechanism that increases the model parameters. Drawing on previous studies, we use a node embedding approach to obtain static information on the graph structure. Through information fusion, GRU coding and similarity calculation, the adjacency matrix is generated that contains static and dynamic information. Finally, with the combination of GCN and GRU, temporal and spatial information can be captured simultaneously. The full model eventually enables accurate traffic prediction.

III. CORRELATION ANALYSIS

Spatial and temporal correlations are crucial factors in achieving accurate predictions. To gain a more intuitive understanding of the spatial-temporal correlations of traffic data, we perform a comprehensive correlation analysis in this section. This analysis will provide a solid theoretical foundation for subsequent studies.

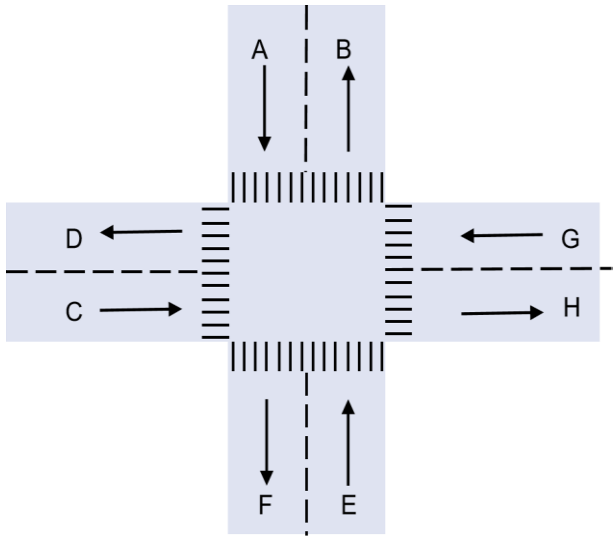


FIGURE 2. Schematic of the lanes at the intersection.

A. TEMPORAL CORRELATION ANALYSIS OF TRAFFIC FLOW

Traffic flow represents a time series data that is affected by various factors such as people's travel habits, holidays, and working hours. It exhibits a periodic character. In Fig. 2, we observe an intersection on a main road in a western Chinese city. Traffic flow detectors on each road are labeled A through H, and arrows in the figure indicate the direction of the road.

Fig.3. illustrates the traffic flow in the section of the road where detector A is located, sampled at intervals of 5 minutes over a two-week span. It is evident that the traffic flow exhibits clear weekly and daily periodic patterns. In addition, distinct traffic patterns can be observed between work days and holidays. However, it is commonly observed that the predicted data exhibit a stronger correlation with the proximity traffic data in the proximity time interval compared to the weekly or daily proximity traffic data. We perform an analysis of traffic flow correlations at neighboring times.

As an example, we consider the traffic flow data collected by detector A on January 8, 2018 as shown in Fig.3. We set T=12, which implies that we analyze the correlation between the current data and the data from the preceding 12 adjacent sampling points (one hour of data). The analysis reveals a robust correlation between the current time and the six neighboring observation points. However, the correlation gradually weakens as the time interval increases. This observation suggests that the variability in traffic flow is significantly influenced by the neighboring traffic flow.

B. SPATIAL CORRELATION ANALYSIS OF TRAFFIC FLOW

To study the spatial correlation of traffic flows, we employ the Pearson correlation coefficient as a measure of the correlation between time series. This coefficient is a commonly used statistic to quantify the strength and direction of a linear relationship between two variables. The formula for calculating

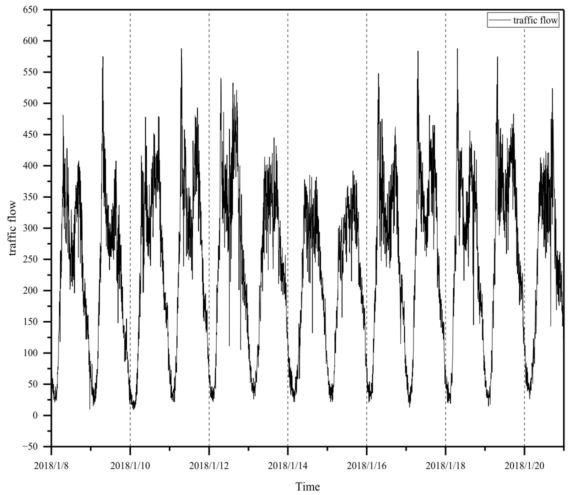


FIGURE 3. **Traffic flow data of detector A over two weeks.**

FIGURE 4. **Spatial correlation coefficient matrix of one day's traffic flow data.**

the Pearson correlation coefficient is as follows [28]:

$$\rho_{XY} = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E\left((X-\mu_X)(Y-\mu_Y)\right)}{\sigma_X \sigma_Y} \tag{1}$$

$\mu_X, \mu_Y, \sigma_X, \sigma_Y$ are the mean and variance of X and Y, respectively.

Traffic flow data from detectors A, B, D, E, and G in Fig. 2 are selected for analysis of spatial correlations over a span of one day. From Fig. 4, it is evident that the correlation coefficient between detectors B and E is 0.98, indicating a high level of correlation between these two detectors. The correlation coefficient between detectors A and G is 0.89, the lowest correlation coefficient.

From the analysis, it has been observed that the strong correlation between detectors B and E can be attributed to their shared traffic direction. In addition, it was found that detector B is located downstream of detector E on the same segment. These factors contribute to the strong correlation between the traffic flow data recorded at detectors B and E. Upon closer analysis, it has been established that the weaker correlation between segments A and B compared to segments B and E is due to the inconsistency in their traffic directions. Although segments A and B are adjacent in spatial location, the traffic flow data collected from these segments exhibit a lower degree of correlation. This can be attributed to the fact that they have different traffic directions, which leads to variations in the traffic patterns and eventually to weaker correlations between the two segments. It has been established that the detectors A and G do have the lowest correlation coefficients. This can be attributed to several factors. First, the road segments in which detectors A and G are located are not adjacent, which means that they may be affected by different traffic conditions and modes. Second, the road directions in these segments are not the same, further contributing to the variability in the traffic flow data. Finally, there is no upstream-downstream relationship between these segments, which affects the consistency of the traffic patterns and leads to weaker correlations. Together, these factors result in the lowest correlation coefficient between detectors A and G. This correlation analysis reveals that the traffic flow at a particular node is influenced by numerous factors, including spatial location and traffic direction. It is crucial to consider these factors, among others, when constructing traffic flow models.

Through the instantiation analysis of spatial-temporal correlations, we have gained a deeper understanding of the relationship between spatial-temporal correlations and influencing factors in traffic data. The analysis results further confirm that in order to achieve accurate traffic prediction, it is necessary to delve into the spatial-temporal correlations of traffic data and fully consider the various factors. More accurate prediction results can only be obtained by considering a comprehensive range of scenarios. Our study focuses precisely on exploring dynamic spatial-temporal relationships in depth.

IV. METHODOLOGY

Before introducing the STDGCN model, we first introduce the notation and basic concepts that will be used in the future.

Traffic forecasting is a multivariate time series forecasting problem, which refers to predicting the future trends of traffic conditions in a certain location or region by analyzing historical data. In general, we define a road network as a graph $G = (V, E, A)$, G represents the graph structure, where V is the set of nodes, $|V| = N$, N is the number of nodes, E is the set of edges, and $A \in R^{N\times N}$ is the adjacency matrix of the graph G [29]. Traffic detectors usually detect multiple features such as flow, vehicle speed, and lane occupancy. We denote the features number by F Then, we use $\mathbf{X} \in R^{T\times N\times F}$ denote a feature tensor on a graph, T represents the input time series length. We use $X_t \in R^{N\times F}$ to denote the d th feature value of the n th node at some time step. $\mathbf{X}$ can be represented by X_t as $\mathbf{X} = \{X_1, X_2, \cdots, X_T\}$.The traffic flow forecasting problem can be defined as (2), predicting the traffic flow in the next time step using the data from the past time slice.

$$[X_{T_P+1}, X_{T_P+2}, \ldots X_{T_p+T_Q}] = f(G; (X_1, X_2, \ldots .X_{T_P})) \tag{2}$$

where T_P represents the historical time steps, T_Q represents the future time step to be predicted and f is the mapping function. The essence of the traffic flow forecasting problem is to find this nonlinear mapping relationship and complete the prediction of future data using historical data.

Fig.5 illustrates the complete structure of STDGCN, which consists of three main components: input encoding module, dynamic adjacency matrix generation module, and prediction module. The input encoding module is responsible for obtaining new encoding vectors that capture valuable spatial-temporal relationships. The dynamic adjacency matrix generation module generates a dynamic adjacency matrix by computing the similarity. Finally, the spatial-temporal relation extraction module combines GCN and GRU to extract spatial-temporal relations and achieve accurate traffic prediction. The structure of STDGCN effectively handles spatial-temporal relationships in traffic data, thus improving the accuracy and efficiency of traffic prediction. Specific implementation methods are detailed in the subsequent sections.

A. INPUT ENCODING MODULE

In traffic prediction, the input information generally includes raw spatial distance information between nodes, node features, and other additional information. Raw spatial distance refers to the straight line distance between road nodes, which can be used to define the adjacency matrix of a road to reflect the connectivity between nodes. Node features include detectable attributes in the traffic network, such as traffic flow, travel speed, lane occupancy, etc. Other additional information includes weather conditions, public events and other factors that affect traffic conditions. By encoding road nodes, we can obtain node embeddings that capture static structural information of the traffic network, especially spatial relational information. By encoding the input features and other feature information, we can obtain their low-dimensional vector representations in the latent space. The encoded information obtained in the initial encoding is fused and fed into the GRU for quadratic encoding, which enables us to obtain hidden states at different time steps containing valuable spatial-temporal information.

Previous studies have shown that it is feasible to construct adjacency matrices through node embeddings to capture static spatial relationships. Hence, we utilize node embedding to deduce the relationship between nodes. According to the traffic road situation, the method of graph embeddings node2vec [30] is used to embed the road structure information into an L-dimensional vector.

Node embeddings are initialized with learnable parameters, randomly, $\mathbf{M} \in R^{N\times \mathrm{L}}$, where each row in M represents a node embedding, N represents the number of nodes, and Lrepresents the dimension of the node embedding. $m_i \in R^L$ represents the node embedding vector of each node.

The input features of nodes can be represented as a $\mathbf{X} \in R^{T\times N\times F}$, where T is the number of time steps, N is the number of nodes, and F is the feature dimension. For other information related to nodes, such as weather conditions or public events, we can encode them for later use. In our study, we focus on the impact of dynamic spatial information on the prediction results. Therefore, at this stage, we do not consider introducing additional information for the time being.

We concatenate the node embeddings vector and the input feature encoding vector at a particular time step to obtain the input to the model as follows: $\mathbf{X}_{in} = [X||M]$, $\mathbf{M} \in R^{N\times \mathrm{L}}$, $\mathbf{M} \in R^{N\times F}$.This input encompasses both static spatial information and dynamic feature information. It is subsequently passed through the GRU for encoding, which helps capture the dependencies between input timesteps. The computation process is represented as:

$$\begin{aligned}
r_t &= \sigma\left(W_r(\mathbf{M}\|\mathbf{X})\right) \\
z_t &= \sigma\left(W_z(\mathbf{M}\|\mathbf{X})\right) \\
\hat{h}_t &= \tanh\left(W_h\mathbf{X} + U_h\left(r_t \odot \mathbf{M}\right)\right) \\
h_t &= (1 - z_t) \odot \mathbf{M} + z_t \odot \hat{h}_t
\end{aligned} \tag{3}$$

W_r, W_z, W_h, and U_h denote weight matrices that require updating, $\odot$ represents the Hadamard product, and *tanh* denotes the activation function. The output h_t contains both static and dynamic information.

B. DYNAMIC ADJACENCY MATRIX GENERATION MODULE

To fully leverage spatial correlations in road traffic, we opt to use GCN to construct our subsequent prediction model. GCN is capable of capturing spatial relationships by aggregating information from neighboring nodes. The amount of information provided by the adjacency matrix directly affects the ability of GCN to extract spatial information. Hence, we show the process of generating the dynamic adjacency matrix in the dynamic adjacency matrix generation module. First, the hidden state at each time step, obtained from the input encoding module, is mapped through a fully connected layer to obtain a vector representation of each node. We denote the transformed vector by V, which is expressed as follows:

$$\mathbf{V} = \text{Fully connected}(h_t), \mathbf{V} \in R^{T\times N}$$

Then, we can get $\mu_{t,i}, \mu_{t,j} \in \mathbf{V}$, $\mu_{t,i}, \mu_{t,j}$ can also be treated as the low dimensional embedding vector for node n_i and n_j at the t-th time step. We exploit cosine similarity [27] to define the dynamic correlations$\beta(n_i, n_j, t)$ between $\mu_{t,i}$ and $\mu_{t,j}$. $\beta(n_i, n_j, t)$ can be calculated as follow:

$$\beta(n_i, n_j, t) = \frac{\mu_{t,i} \cdot \mu_{t,j}}{\left\|\mu_{t,i}\right\| \cdot \left\|\mu_{t,j}\right\|} \tag{4}$$

β reflects the relations between nodes and contains both static and dynamic information on the graph. Our goal in solving β is to use it to represent the dynamic adjacency matrix.The dynamic adjacency matrix A_d can be calculated by

$$A_d[i,j] = \frac{1}{T_P}\sum\nolimits_{\mathrm{t}} \beta(n_i, n_j, t) \tag{5}$$

T_P represents the historical time steps.

The dynamic adjacency matrix is constructed by integrating various information codes. In contrast to the static adjacency matrix, which is defined based on distance, A_d incorporates both spatial and temporal information between nodes over time. This abundant information is valuable for predicting future traffic patterns.

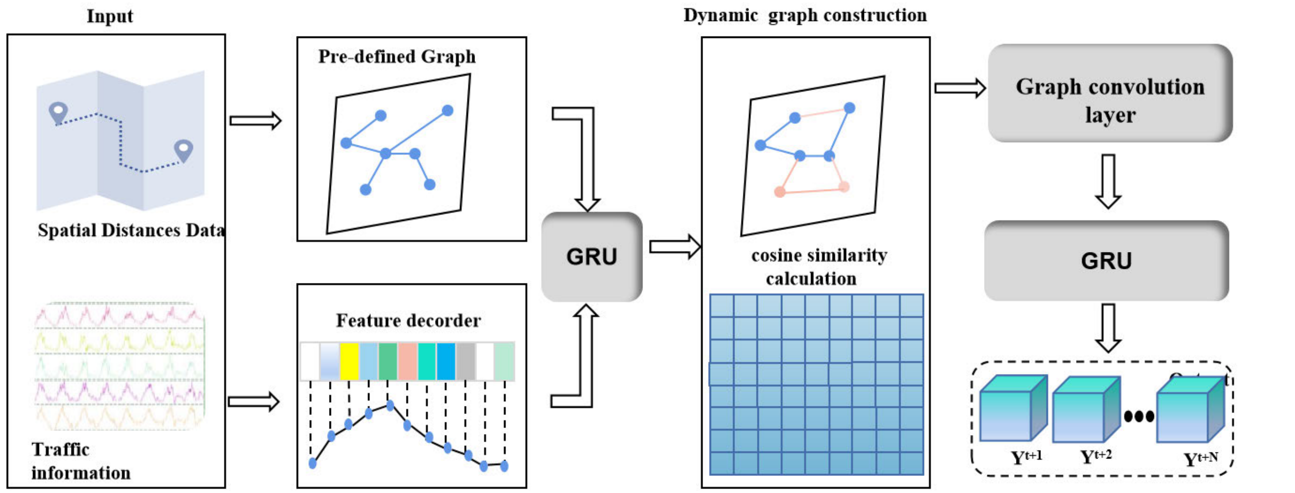


FIGURE 5. Network architecture of STDGCN.

C. SPATIAL-TEMPORAL RELATION EXTRACTION MODULE

In this section, we replace the adjacency matrix in the original GCN model with the dynamic adjacency matrix generated in the previous step. The output of the GCN is then fed into the GRU model. By rewriting the GRU, we enhance the model's ability to capture both temporal and spatial features. Ultimately, this leads to accurate predictions.

We use a GCN with two layers to learn spatial features from traffic data.

$$F(A_{\mathrm{d}}, X_t) = \sigma(\hat{A}_d Relu(\hat{A}_d X_t W_0) W_1) \tag{6}$$

where X_t represents the feature matrix,A_d is dynamic adjacency matrix. $\hat{A}_d = \tilde{D}^{-\frac{1}{2}} \tilde{A}_d \tilde{D}^{-\frac{1}{2}}$ represent the normalization process.$\tilde{A}_d = A_d + I_N, I_N$ is the identity matrix. D is the degree matrix. W_0 and W_1 are weight matrix.

We used GCN to modify GRU in order to obtain both temporal and spatial correlations. The modified expression for GRU is as follows:

$$\begin{aligned} r_t &= \sigma(W_r[g_{*\theta}, h_{t-1}] + b_r) \\ u_t &= \sigma(W_u[g_{*\theta}, h_{t-1}] + b_u) \\ c_t &= tanh(W_c[g_{*\theta}, (r_t \odot h_{t-1})] + b_c) \\ h_t &= u_t \odot h_{t-1} + (1 - u_t) \odot c_t \end{aligned} \tag{7}$$

$g_{*\theta}$ represents the graph convolution process, this is equivalent to (6).

Following the aforementioned stages, we have successfully constructed the STDGCN network. This network is capable of capturing the temporal, geographical, and spatial-temporal relationships present in the provided traffic data. In the experimental phase, we will assess the effectiveness of the network using real-world traffic statistics.

V. EXPERIMENTS

A. DATA DESCRIPTION

In this section, we aim to evaluate the effectiveness of our model using two publicly available datasets. The first dataset, METR_LA, consists of traffic speed data collected using the Loop Detector from 207 sensors located on freeways in Los Angeles County. The data are recorded at 5-minute intervals [22]. The second dataset, PEMSD4, contains flow, occupancy, and speed data from 307 detectors placed on California highways. The data spans 59 consecutive days, starting from January 1, 2018, and is recorded at 5-minute intervals [22]. Although the PEMSD4 dataset provides three types of traffic features, we have chosen to focus on traffic speed as the prediction objective in both datasets. This decision was made to ensure fairness in evaluating model performance and to provide a clear basis for comparative analysis. By using the same features for prediction, we can accurately compare the performance of the models on different datasets and eliminate any interference caused by differences in feature selection.

TABLE 1. Datasets-specific information.

Datasets	PEMSD4	METR_LA
Location	Seattle Area, USA	Los Angeles County, USA
Number of Sensors	307	207
Sample interval	Every 5 minutes	Every 5 minutes
Characteristics	Flow, occupy, speed	Vehicle speed
Total amount of data	Jan. 1, 2018 - Dec.31, 2015	Mar. 1, 2012 - Jun. 27, 2012

In TABLE 1, we provide the details of the datasets. We use historical speed data from the past hour to predict the traffic speed for the next hour. Specifically, we use the data from the past 12 time steps to predict the data for the next 12 time steps. It is important to note that our model can also be applied to predict other types of traffic metrics, such as traffic volume and road occupancy. By utilizing historical data for prediction, we can provide valuable insights about future traffic conditions.

In subsequent trials, we split the dataset into three sections: training set, validation set, and test set, with a ratio of 0.7:0.1:0.2. This partition allows us to train the model on a large portion of the data, validate its performance on a separate set, and finally evaluate its generalization ability on unseen data in the test set.

B. EVALUATION METRICS

To evaluate the effectiveness of the model, we selected mean absolute error (MAE) and root mean square error (RMSE) as performance assessment metrics. The calculation formulas for these metrics are as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$
$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{n}}\sum(y_i - \hat{y}_i)^2} \tag{8}$$

where y_i and $\hat{y}_i$ denote the predicted and true traffic flow or speed at the i-th point in time t, respectively. Lower MAE and RMSE values, in general, indicate higher performance of the model in predicting the target variable [31].

As the loss function, we used mean squared error (MSE).

$$\mathrm{Loss} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{9}$$

C. PARAMETERS SETTING

We implemented the proposed model using PyTorch 1.9.0 and performed all experiments on a cloud server. The server was equipped with an Intel(R) Xeon (R) Platinum 8358P CPU @ 2.60GHz, an RTX 3090 (24GB) GPU, and 80GB of memory. During the training process, we used the Adam optimizer with a learning rate of 0.001. The model was trained with a batch size of 64. To prevent overfitting, we incorporated early stopping in the training procedure.

D. BASELINES

HA: Historical average models are the most basic statistical-based models that use statistical data from a large number of historical eras as observations. These models compute the arithmetic mean of the historical data and use it as a prediction value for the following time period.

FNN: A feed-forward neural network with three hidden layers has been created and used to evaluate the performance of a simple neural network in traffic prediction.

T-GCN: Spatial correlations are captured by the GCN, while temporal correlation links are obtained by the GRU using the GCN computation results.

DCRNN: The traffic process is viewed as a graph-based diffusion process. It uses an upgraded GRU to capture temporal correlations and a diffusion convolutional neural network to capture spatial correlations.

Graph WaveNet: It captures spatial correlations through a constructed adaptive adjacency matrix and a predefined graph structure, and temporal correlations are obtained using diffusion causal convolution.

A**STGCN**: Spatial-temporal graph convolutional networks based on attention mechanisms establish a temporal-based attention mechanism and a spatial-based attention mechanism, respectively, to further extract temporal and spatial correlations.

AGCRN: The innovation of the adaptive graph convolution recursive model lies in learning the adaptive parameters of each node through the Node Adaptive Parameter Learning (NAPL) and Data Adaptive Graph Generation (DAGG) modules, and ultimately obtaining the adaptive adjacency matrix.

E. RESULTS AND DISCUSSION

We conducted comparative experiments between STDGCN and seven baseline models on the METR_LA and PEMSD4 datasets. In the experiment, we use the data from the past 12 time steps to predict the traffic speed values for the next 12 time steps. TABLE 2 shows the errors of the different models in predicting future values of the 3rd, 6th and 12th time steps on the two datasets, that is, the errors in predicting traffic speed values at 15, 30 and 60 minutes. These values will be used to evaluate the performance of the model.

Analyzing the results in the table carefully, we can draw the following conclusions:

Of all the approaches, the conventional statistical method HA performs the lowest. The main reason for this is that HA ignores geographical relationships in the traffic situation. It relies solely on historical data obtained through sliding windows, which limits the collection to only temporally relevant data. Moreover, we observe that the predictions generated by HA remain unchanged when the prediction range is enlarged. This suggests that HA lacks the ability to capture and incorporate spatial dependencies and fails to adapt to longer-term forecasting scenarios

FNN, as a basic neural network technique, has the capability to simulate more complex and nonlinear traffic predictions. It performs better compared to conventional prediction methods. However, FNN is limited in its ability to extract only time-dependent relationships from traffic data. The necessity of propagating the data for long-term prediction also results in a lack of sequential links between the data, which ultimately reduces the accuracy of the prediction. While FNN may be effective for short-term prediction, it may struggle to capture long-term patterns and dependencies in the data.

T-GCN, DCRNN, Graph WaveNet, ASTGCN and AGCRN are all GCN based models specifically designed for traffic prediction. As an excellent baseline model, it achieves excellent performance on our two selected datasets. These models utilize a graph structure to represent road traffic networks, which is more suitable to capture complex relationships and dependencies in traffic data. DCRNN combines traffic flow and diffusion process to model spatial dependencies, which improves the ability of the model to obtain spatial information. Graph WaveNet combines the WaveNet structure with an adaptive adjacency matrix, which aims to adaptively update the adjacency matrix and mine spatial correlations at a deeper level. ASTGCN incorporates an attention mechanism to explore spatio-temporal correlations in the data more comprehensively. On the other hand, AGCRN utilizes an adaptive adjacency matrix, which obtains a dynamic adjacency matrix by updating

TABLE 2. Performance of different models on METR_LA and PEMSD4 datasets.

MODEL	METR_LA(15/30/60min)		PEMSD4(15/30/60min)	
	MAE	RMSE	MAE	RMSE
HA	10.92/10.92/10.92	22.42/22.40/22.40	12.44/12.44/12.44	25.05/25.05/25.05
FNN	5.70 /6.10 /8.49	8.12 /10.54/13.52	9.69/10.58 /12.82	21.59/23.72/28.60
T-GCN	4.76/ 5.07 /7.03	7.79 / 9.59 /11.67	7.69 / 9.17 / 12.33	19.54/22.91/29.99
DCRNN	5.36/ 6.84 / 6.82	7.34/ 8.78 / 10.22	6.87 / 7.57 / 8.76	17.34/18.75/21.22
GraphWaveNet	5.36/ 6.84 / 6.81	5.42 / 6.42 / 7.42	7.36 / 7.84 / 8.82	18.04/19.04/20.89
ASTGCN	5.22/ 6.54/ 8.18	8.03 / 9.28 /10.69	7.61 / 8.03 / 8.85	18.56/19.55/21.21
AGCRN	9.33/ 9.88/ 10.37	5.48 / 6.54 / 7.50	6.64 / 6.99 / 7.48	17.19/18.04/19.17
Our model (STDGCN）	**4.56 / 5.02 / 6.91**	**5.45 / 6.41 / 7.45**	**6.53 / 6.80/ 7.36**	**17.06/18.01/19.08**

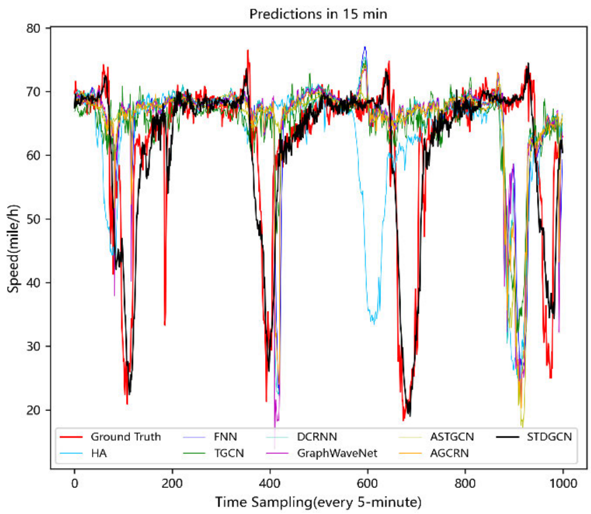


FIGURE 6. Prediction visualization in 15 min.

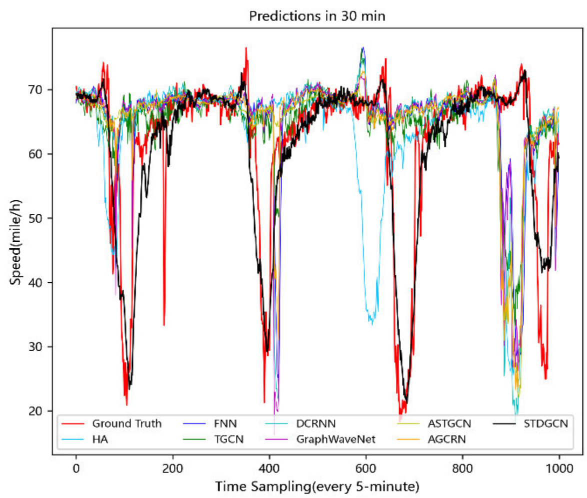


FIGURE 7. Prediction visualization in 30 min.

node parameters. Through analysis, it has been observed that the aforementioned models place greater emphasis on mining spatial correlations of traffic information. As a result, their prediction results outperform conventional prediction methods and simple FNN models.

On the METR_LA dataset, the MAE values of baseline models when predicting the traffic speed in the future 15, 30, and 60 minutes are 4.76, 5.07 and 6.81, respectively. The MAE of our STDGCN model at the corresponding time steps is 4.56, 5.02, and 6.91. These values are 4.20%, 0.99%, and 1.44% higher than the optimal value achieved by the baseline model.

In the 15, 30, and 60 minutes predictions, the AGCRN model from the baseline models achieves the lowest RMSE values of 5.48, 6.54, and 7.50, respectively. In comparison, our model has RMSE values that are 0.55%, 1.99%, and 0.67% higher than the AGCRN model, respectively. Looking at the data in Table 2, we find that our model performance also improves to varying degrees on the PEMSD4 dataset.

STDGCN model is shown to have certain advantages in capturing spatio-temporal correlations. Our proposed method for generating dynamic adjacency matrix has shown superior performance compared to models that directly utilize predefined graph structures for prediction. By integrating dynamic feature extraction and node embedding, our method comprehensively considers the impact of spatial correlation between nodes in the graph. Additionally, the calculation of similarity enables us to obtain the spatial similarity relationship between nodes, which provides better support for subsequent prediction tasks. For a more intuitive understanding, we visualize the prediction results for the second node in the PEMSD4 dataset at different time steps in Fig. 6, Fig.7, and Fig.8.

The core concept of STDGCN model is to incorporate a dynamic adjacency matrix, which enables spatial relationships between nodes to evolve over time. This dynamic adjacency matrix is computed based on the temporal intervals and spatial distances of the traffic data and updated at each

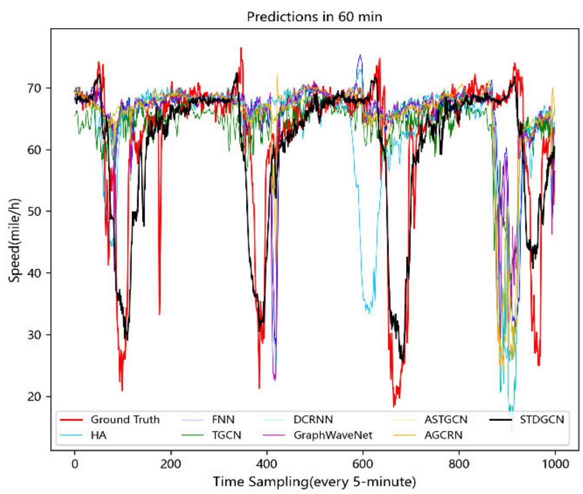


FIGURE 8. Prediction visualization in 60 min.

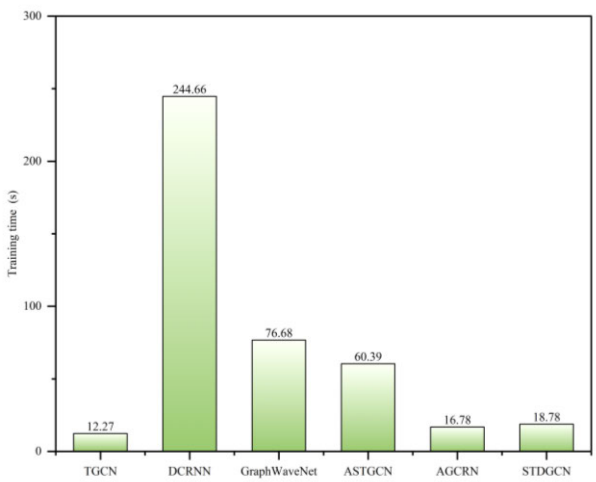


FIGURE 9. Training time comparison.

time step. As a result, the model is able to effectively capture the dynamic spatial relationships between nodes at different time steps.

To further investigate the impact of each component of the STDGCN model on its performance, ablation experiments were performed. The basic modules responsible for the generation of the dynamic adjacency matrix, namely the GRU module of the input encoding module and the cosine similarity computation module, are removed. Table 3 shows the performance metrics of the model for predicting 30 minutes of data under different module deletion scenarios.

By comparison, it can be observed that removing the essential module for generating the dynamic adjacency matrix significantly deteriorates the performance of the model. This is because the GRU coding module captures both static spatial information and dynamic temporal features. Meanwhile, the cosine similarity computation module updates the temporal-spatial correlations between nodes at each time step, enabling the generation of an adjacency matrix that deeply explores the temporal and spatial dependencies between nodes.

TABLE 3. Comparison results between static and dynamic adjacency matrices.

Removed modules	METR_LA (30min)	
	MAE	RMSE
GRU module（In the input encoding module）	8.66	18.96
Cosine similarity calculation module	8.82	18.01

Thus, the proposed method in STDGCN model effectively incorporates both static and dynamic spatiotemporal features and further leverages similarity computation to enhance prediction accuracy.

In addition to the aforementioned experimental content, we compare the training time of the five baseline models with the STDGCN model to assess the operational efficiency of the model and its demand on computational resources. Specifically, we compare the average computation time required to train each model for one epoch. The results of this comparison are shown in Fig. 9. We observe that the DCRNN model requires the longest training time among the aforementioned models, with a duration of 244.66 seconds per epoch. T-GCN, Graph Wavenet, ASTGCN and AGCRN models are trained for 12.27, 76.68, 60.39 and 16.78 seconds per epoch, respectively. The training time of STDGCN is 18.78 seconds per epoch. Although the training time of STDGCN is slightly longer than that of TGCN and AGCRN, the prediction accuracy of STDGCN is better than that of the previous experiments. Therefore, the longer training time of STDGCN can be considered as a trade-off for its improved prediction accuracy, compensating for the time consumption.

VI. CONCLUSION AND FUTURE WORK

We propose a spatial-temporal dynamic graph convolutional neural network for the traffic prediction task. With an input encoding module, we extract hidden states at each time step that contain both static spatial and dynamic temporal information. Using cosine similarity computation, we generate a dynamic adjacency matrix that captures dynamic spatial-temporal information. By combining GCN and GRU, we leverage this dynamic adjacency matrix for the traffic prediction task. Through experiments on two datasets, we validate the feasibility of the proposed method for generating dynamic adjacency matrices and further demonstrate the significant impact of extracting dynamic spatial relationships on the accuracy of traffic prediction. In future research, we will continue to explore the relationship between temporal, spatial and spatial-temporal correlations and consider external factors such as weather to further enhance the performance of multidimensional and multi-step traffic prediction models.

REFERENCES

[1] Z. Chen, Z. Lu, Q. Chen, H. Zhong, Y. Zhang, J. Xue, and C. Wu, "Spatial–temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism," *Inf. Sci.*, vol. 611, pp. 522–539, Sep. 2022, doi: 10.1016/j.ins.2022.08.080.

[2] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3634–3640.

[3] W. Zhong, H. Meidani, and J. Macfarlane, "Attention-based spatial–temporal graph neural ODE for traffic prediction," 2023, *arXiv:2305.00985*.

[4] D. V. Phuong and T. M. Phuong, "Gender prediction using browsing history," in *Proc. 5th Int. Conf. Knowl. Syst. Eng. (KSE)*. Hanoi, Vietnam: Springer-Verlag, vol. 244, Oct. 2013, pp. 271–283, doi: 10.1007/978-3-319-02741-8_24.

[5] F. Huang, P. Yi, J. Wang, M. Li, J. Peng, and X. Xiong, "A dynamical spatial–temporal graph neural network for traffic demand prediction," *Inf. Sci.*, vol. 594, pp. 286–304, May 2022, doi: 10.1016/j.ins.2022.02.031.

[6] G. O. Ferreira, C. Ravazzi, F. Dabbene, G. C. Calafiore, and M. Fiore, "Forecasting network traffic: A survey and tutorial with open-source comparative evaluation," *IEEE Access*, vol. 11, pp. 6018–6044, 2023, doi: 10.1109/access.2023.3236261.

[7] G. Zheng, W. K. Chai, and V. Katos, "A dynamic spatial–temporal deep learning framework for traffic speed prediction on large-scale road networks," *Exp. Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116585, doi: 10.1016/j.eswa.2022.116585.

[8] L. Yang, Q. Yang, Y. Li, and Y. Feng, "K-nearest neighbor model based short-term traffic flow prediction method," in *Proc. 18th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. (DCABES)*, Nov. 2019, pp. 27–30, doi: 10.1109/DCABES48411.2019.00014.

[9] T. H. H. Aldhyani and M. R. Joshi, "Enhancement of single moving average time series model using rough k-means for prediction of network traffic," *Int. J. Eng. Res. Appl.*, vol. 7, no. 3, pp. 45–51, Mar. 2017, doi: 10.9790/9622-0703064551.

[10] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A, Transp. Sci.*, vol. 15, no. 2, pp. 1688–1711, Nov. 2019, doi: 10.1080/23249935.2019.1637966.

[11] F. Li, J. Feng, H. Yan, G. Jin, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," 2021, *arXiv:2104.14917*.

[12] L. Mou, P. Zhao, H. Xie, and Y. Chen, "T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction," *IEEE Access*, vol. 7, pp. 98053–98060, 2019, doi: 10.1109/access.2019.2929692.

[13] H. Li, S. Yang, Y. Song, Y. Luo, J. Li, and T. Zhou, "Spatial dynamic graph convolutional network for traffic flow forecasting," *Appl. Intell.*, vol. 53, pp. 14986–14998, Nov. 2022, doi: 10.1007/s10489-022-04271-z.

[14] M. Owais, G. S. Moussa, and K. F. Hussain, "Robust deep learning architecture for traffic flow estimation from a subset of link sensors," *J. Transp. Eng., A, Syst.*, vol. 146, no. 1, p. 290, Jan. 2020, doi: 10.1061/jtepbs.0000290.

[15] A. Alshehri, M. Owais, J. Gyani, M. H. Aljarbou, and S. Alsulamy, "Residual neural networks for origin–destination trip matrix estimation from traffic sensor information," *Sustainability*, vol. 15, no. 13, p. 9881, Jun. 2023, doi: 10.3390/su15139881.

[16] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," 2021, *arXiv:2101.11174*.

[17] Y. Chen, K. Li, C. K. Yeo, and K. Li, "Traffic forecasting with graph spatial–temporal position recurrent network," *Neural Netw.*, vol. 162, pp. 340–349, May 2023, doi: 10.1016/j.neunet.2023.03.009.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[19] J. Bi, X. Zhang, H. Yuan, J. Zhang, and M. Zhou, "A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 3, pp. 1869–1879, Jul. 2022, doi: 10.1109/TASE.2021.3077537.

[20] Z. Li, J. Yu, G. Zhang, and L. Xu, "Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time series forecasting," *Exp. Syst. Appl.*, vol. 216, Apr. 2023, Art. no. 119374, doi: 10.1016/j.eswa.2022.119374.

[21] W. Li, X. Wang, Y. Zhang, and Q. Wu, "Traffic flow prediction over muti-sensor data correlation with graph convolution network," *Neurocomputing*, vol. 427, pp. 50–63, Feb. 2021, doi: 10.1016/j.neucom.2020.11.032.

[22] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[23] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," 2019, *arXiv:1906.00121*.

[24] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. Nat. Conf. Artif. Intell.*, 2019, pp. 1–9, doi: 10.1609/AAAI.V33I01.3301922.

[25] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 914–921, doi: 10.1609/aaai.v34i01.5438.

[26] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," 2020, *arXiv:2007.02842*.

[27] G. Luo, H. Zhang, Q. Yuan, J. Li, and F.-Y. Wang, "ESTNet: Embedded spatial–temporal network for modeling traffic flow dynamics," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19201–19212, Oct. 2022, doi: 10.1109/tits.2022.3167019.

[28] B. Lou, C. Li, J. Deng, L. Zhu, C. Yang, and W. Chen, "Low-voltage distribution network topology verification method based on revised Pearson correlation coefficient," *J. Phys., Conf. Ser.*, vol. 1633, no. 1, pp. 012084–012091, 2020, doi: 10.1088/1742-6596/1633/1/012084.

[29] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," 2019, *arXiv:1911.08415*.

[30] A. Grover, J. Leskovec, and M. Assoc Comp, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, San Francisco, CA, USA: Association for Computing Machinery, Sep. 2016, pp. 855–864, doi: 10.1145/2939672.2939754.

[31] W. J. Xiao and J. F. Liu, "Research on traffic flow forecasting method based on graph neural network," in *Proc. 2nd IEEE Int. Conf. Softw. Eng. Artif. Intell. (SEAI)*. Quanzhou, China: Huaqiao University, Jun. 2022, pp. 243–247, doi: 10.1109/seai55746.2022.9832353.

WENJUAN XIAO received the master's degree from the College of Electrical and Information Engineering, Lanzhou University of Technology, where she is currently pursuing the Ph.D. degree with the College of Electrical and Information Engineering. She is also a Lecturer with Northwest Minzu University. Her research interests include machine learning, deep learning, and intelligent transportation systems. She has presided over one Natural Science Foundation of Gansu Province and three fundamental research funds for the central universities.

XIAOMING WANG is currently a Professor and a Ph.D. Supervisor with the Lanzhou University of Technology. He has won the first prize for teaching achievements in Gansu Province, the Gansu Science and Technology Progress Award, and many other teaching and scientific research achievements. He has published dozens of teaching and scientific research articles. His research interests include smart transportation systems and rail transportation systems.

Industrial Engineering and Applications
L.-C. Tang (Ed.)

doi:10.3233/ATDE230068

Research on Time Series Imputation Based on Generative Adversarial Network

Wenjuan Xiao[a,b,1] and Hui Lv[b]
[a] *College of Electrical and Information Engineering, Lanzhou University of Technology*
[b] *College of Electrical Engineering, Northwest MinZu University*

Abstract. Time series data is collected in chronological order to represent how the collected data changes over time. This type of data is susceptible to interference from external factors that ultimately make the data missing. Missing data will cause the lack of some historical information, and it is not conducive to the development of downstream tasks. In recent years, Generative Adversarial Networks(GAN) GAN is widely used in image processing tasks and has achieved good results ,which can also be applied to time series generation and interpolation tasks. We analyze the types of time series data missing, the structure of general GAN, and compare three models such as E2GAN, BIGAN and MBGAN, focusing on the comparison of the model composition of generator and discriminator, so as to provide ideas for the subsequent optimization application of GAN in time series data imputation.

Keywords. Imputation, GAN, Data missing, Generator, Discriminator, Time series data.

1. Introduction

Time series data is the most widely used data in daily life. Such as temperature data, stock data, traffic flow data, patient physiological index monitoring data, etc. These data provide the reference basis for daily life, travel, employment and so on. However, the acquisition of time series data requires the relevant equipment to work continuously and normally, so the data is easily disturbed by external factors such as sudden power cut, machine error and so on. Missing data will cause the loss of effective information, which is not conducive to capturing the temporal relationship and potential connection among the data in the complete observation cycle, and will bring trouble to the development of subsequent tasks such as prediction and evaluation. Data imputation is to fill the missing data caused by external reasons by mining the internal relationship between the original data and using relevant algorithms. The imputed data should be the same distribution as the original data.

Missing data can be classified as missing completely at random(MCAR), missing at random(MAR), missing not at random(MNAR)[1].MCAR means that the missing data is random, and the missing data does not depend on other variables. MAR refers to missing data dependent on other complete variables. MNAR means that the absence of data depends on the incomplete variable itself. Missing data processing can be basically classified into three methods: deletion method, the imputation method based on

[1] Corresponding Author: Wenjuan Xiao, Contact details.xwj@xbmu.edu.cn

statistics, and the imputation method based on machine learning. The direct deletion method aims at the time series that is completely missing at random. At the same time, only when the amount of missing data is small and the impact on subsequent tasks is small, this part of missing values can be ignored or deleted directly. This method is bound to lose important information in the data, and is not suitable for tasks where the data single node contains rich information.

Statistics-based algorithms usually follow a strict stationarity assumption, that is, assume that the data changes follow a probability distribution, and use the data values that best matches the predicted probability distribution to impute the missing data points[2]. Statistical method imputation includes adjacency imputation, eigenvalue imputation and linear interpolation[3]. Based on machine learning imputation methods, common methods include nearest neighbor method (KNN), recurrent neural Network (RNN), random forest and matrix factorization based missing values imputation algorithms[4,5]. These methods need to mine the internal relationship between data and are more suitable for random missing data.

With the rapid development of generative adversarial networks(GAN) and its excellent performance in the image generation task, many experts and scholars have applied GAN to the generation and interpolation of time series data, and achieved good results[6,7]. The essence of these methods is to use the data near the missing point as the feature, predict the data of the missing point, and mine similar change models from a large number of historical data, so as to carry out more accurate data filling. E^2GAN[8], BIGAN[9,10] and MBGAN[11] are typical models for imputation and filling of time series data using GAN, and we will further compare and analyze in the following.

2. Basic Definition of GAN

Generative Adversarial Networks (GAN) is a kind of generative neural network proposed by Goodfellow[12] et al in 2014. The structure of the generative adversarial network is composed of a generator and a discriminator. During the training process, the task of the generator G is to continuously generate data (fake data), and the first input of the generator is random noise. The task of the discriminator is to determine whether the result generated by G is true or false. The two networks fight against each other. The generator G tries to generate fake data that can deceive D, and the discriminator D tries to identify the fake data generated by G. Through continuous training, when the probability of D identifying fake data reaches 50%, the balance is reached and the generator achieves good results.The basic structural framework of GAN is shown in Fig. 1.

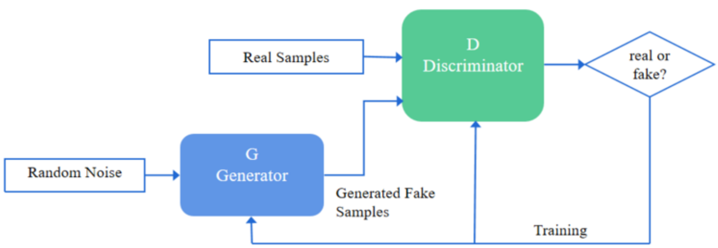


Figure 1. Basic Structure of GAN

The goal of GAN can be expressed as Eq.1:

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))] \tag{1}$$

The optimization of the discriminator is realized by implementing $\max_D V(D,G)$ .V(D,G) is the objective function of the discriminator, where $E_{x\sim P_{data}(x)}[\log D(x)]$ represents the mathematical expectation of the discriminator on the probability of discriminating a sample from the real data distribution. For real data, the probability of predicting a positive sample is closer to 1, the better the effect is. $E_{z\sim P_z(z)}[\log(1-D(G(z)))]$ represents the expectation of the negative logarithm of the prediction probability obtained after the sample sampled from the noise data distribution is sent to the discriminator through the generator to generate data. Optimizations of the generator pass Implementation $\min_G \max_D V(D,G)$.

3. Preliminary

In this section, we will briefly introduce some basic definitions and notations that will be used to fill missing data. Time series data in daily life are divided into univariate time series data and multivariate time series data. Univariate and multivariate are distinguished by the number of characteristic variables at a certain time observation point. For example, the weather data of a specific day is actually composed of multiple features such as temperature, humidity, and air quality of the day, so the collection of weather data is multivariate time series data. This paper focuses on the imputation of missing values for multivariate time series data.

We denote multivariate time series dataset contains a set of samples $S=\{X_1,X_2,\cdots,X_s\}$.Each sample in S,i.e., $X=\{x_{t_1},x_{t_2},\cdots,x_{t_n}\}^T$ is a time series matrix observed at the timestamp lists $T=(t_1,t_2,\cdots,t_n)$ .In particular,the *i-th* observation of X *is* x_{t_i} ,which consists of d attributes $\{x_{t_i}^1,x_{t_i}^2,\cdots,x_{t_i}^d\}$.

We use mask matrix M to denote the missing values. $M\in R^{n\times d}$. And each element of M can be defined as Eq.2:

$$M_{t_i}^j = \begin{cases} 0 & \text{if } x_{t_i}^j \text{ is missing} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

Because the sampling interval of time series is not necessarily fixed, in order to record the time difference between two adjacent data in time series, δ matrix is introduced to record the observation time difference between the current value and the last effective value, which is convenient for the subsequent calculation of the decay factor of historical memory vector. The followings is the calculation way of δ .

$$\delta_{t_i}^{j} = \begin{cases} t_i - t_{i-1}, & M_{t_i-1}^{j} = 1 \\ \delta_{t_i-1}^{j} + t_i - t_{i-1}, M_{t_i-1}^{j-1} == 0\& > 0 \\ 0, & i == 0 \end{cases} \tag{3}$$

There is an example to illustrate the relationship between X , T, M and δ .In the following example, d=4, n=4 and "none" is missing value.

$$X = \begin{bmatrix} 3 & 5 & none & 1 \\ 2 & none & 7 & 3 \\ 8 & 9 & 2 & none \\ none & 2 & none & 2 \end{bmatrix} \tag{4}$$

$$T = \begin{bmatrix} 0 \\ 5 \\ 12 \\ 16 \end{bmatrix} \tag{5}$$

$$M = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{6}$$

$$\delta = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 \\ 7 & 12 & 7 & 7 \\ 4 & 4 & 4 & 11 \end{bmatrix} \tag{7}$$

4. Method

In order to systematically study the application effect of GAN in time series data imputation, we want to mine the similarities and differences between models. We selected three models such as E^2GAN, BIGAN and MBGAN for comparative analysis. The reason that we choose these three models is that they all take the generative adversarial network as the basic structure, and perform different degrees of improvement in the generator and discriminator. They are combined the deformation of RNN, such as LSTM, GRU and other structures to complete tasks such as filling and forecasting of time series data. In the following, we will carry out further comparative analysis of these three models.

4.1. E^2GAN

Luo et al.,proposed E^2GAN in 2019. It is a further application of Gated Recurrent Units for Imputation (GRUI)[8]. GRUI decays the historical memory vector of this dimension according to the length of the data missing time, that is, the δ matrix. To accurately reflect the effect of the duration of missing data on the current observation value, the attenuation factor β is introduced, and the formula is expressed as follows:

$$\beta_{t_i} = 1 \Big/ e^{max(0, w_\beta \delta_{t_i} + b_\beta)} \tag{8}$$

Where, $\beta \in (0,1]$ are the training parameters, denotes the time interval of each dimension at the current moment. The decay of the historical memory vector is inversely proportional to the growth of the time interval. The larger the time interval is, the smaller the influence of the historical memory vector on the current imputation data is. The updated GRU and GRUI are shown in FIG. 2.

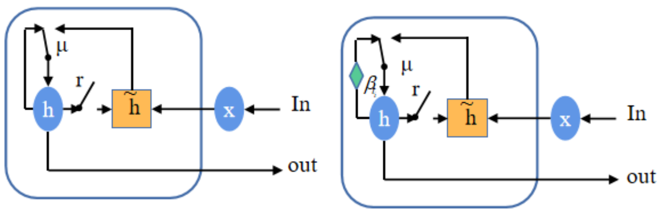


Figure 2. (a) GRU (b)GRUI

The overall framework of E^2GAN is based on GAN, contains a generator and discriminator, and adopts the idea of binary game for adversarial training. The overall framework of the E^2GAN is shown in Fig.3.

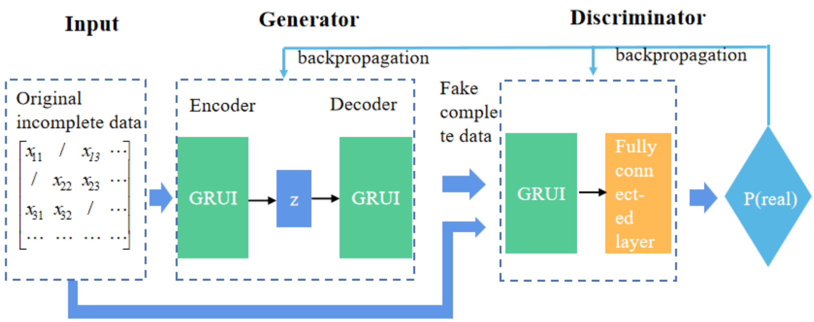


Figure 3. E^2GAN architecture overview

The G is constructed by Encoder-Decoder. The input of encoder part is a time series data containing missing values. After compression by the denoising autoencoder containing GRUI, the low-dimensional feature expression vector z of the input data can be obtained. In the decoder part, the GRUI unit is also used to reconstruct the complete time series data. After multiple rounds of training, the generated data that can fill the original data is obtained. This means that the generator is able to generate new samples based on the original data x that conform to the distribution of the original dataset.

The input of D consists of two parts, the real missing time series data and the complete fake time series data generated by the G. The main component of the D is still the recurrent neural unit, whose main role is to process time series to obtain its historical memory vector. The output part is one-dimensional mapping by a fully connected layer neural network, and finally the sigmoid function is used to compress the output value. At the same time, in order to solve the mode collapse problem, the algorithm uses WGAN with relatively more stable training. The goal of WGAN can be expressed as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data}(x)}D(x) + E_{z\sim P_z(z)}[-D(G(z))] \tag{9}$$

E²GAN is suitable for multivariate time series filling and achieves good imputation results.

4.2. BIGAN

BIGAN is a time series imputation model proposed by Mehak Gupta et al in 2020.The main structure of this model is still the GAN[9]. LSTM is the main method adopted in the G module and the D module.

G uses a bidirectional recurrent network, and the forward training and backward training generate a value respectively. After multiplying the two values by their respective impact factors for missing data, they are sent to the fully connected layer to generate the filling value of the missing position.D is also relatively simple and consists of Bidirectional LSTM .D uses the cross-entropy loss function to improve its ability to distinguish between the true value and the generated value through continuous training.The specific structure of BIGAN is shown in Fig.4.

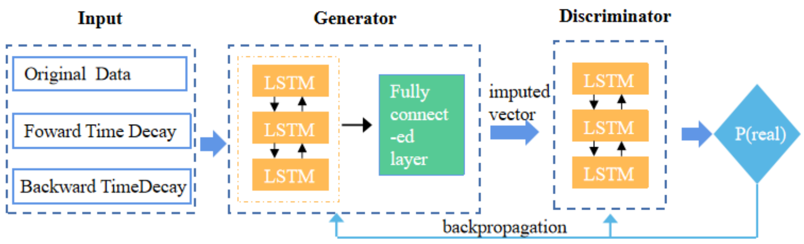


Figure 4. BIGAN architecture overview

The BIGAN model employs four different loss functions. The generator loss is composed of the sum of the mask reconstruction loss function, the classification loss function of the generator, and the consistency loss function of the difference between the forward generation value and the backward generation value. The cross-entropy loss function is used in the discriminator part.

The model can be used for data imputation and subsequent prediction task. In the imputation setting, in addition to the original missing data, part of the data should be randomly deleted. In the end, the original missing data and deleted data should be filled. In the prediction task, the time series is divided into observation window and prediction window with missing value, and all the window values are predicted. The model is verified by three real data sets, and filling effect is good.

4.3. MBGAN

MBGAN is a time series imputation model proposed by Ni etal in 2020[11]. The main framework of this model is still GAN. This model pays more attention to the mutual influence between multivariate time series data, so before sending the data into the generation module, the Decision Tree and SVM- RFE are used to extract the features of the data.Fig.5 is the structural block diagram of MBGAN.

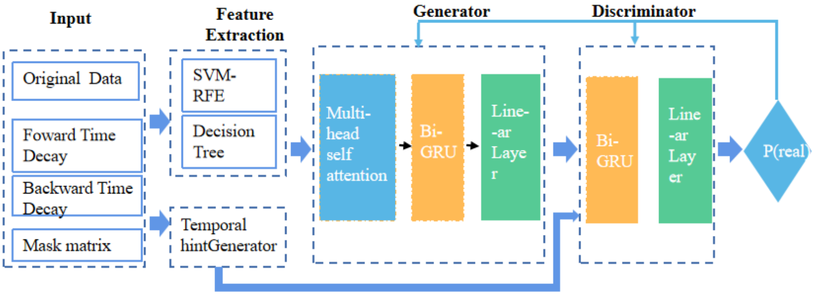


Figure 5. MBGAN architecture overview

The overall inputs to MBGAN are time series data with missing values, a mask matrix, forward time decay and backward time decay time series data. The forward time decay and backward time decay are borrowed from the BIGAN model.

In order to deeply mine the relationships between multivariate input features, a multi-head self-attention mechanism is introduced into the generator, and a bidirectional GRU is used to reveal the dependencies between time series data. Finally, the data is generated through the linear layer output. G is also discriminated by a bidirectional GRU layer and a linear regression layer.

5. Experment

5.1. Datasets

In order to compare the performance of the three models, we verify them through experiments. We use an air quality datasets of a Chinese city including an hourly data from May 2014 to May 2015. The dataset contains more than 8000 data, each containing six features related to weather conditions :pm2.5, pm10, SO2, NO2, CO and O3. To facilitate comparative analysis, pm2.5 features are selected for repair.

5.2. Evaluation Metrics

We randomly deleted between 10 % and 30% of the data, simulating missing data completely at random. In this work, we use two commonly used metric, MAE and RMSE. Specifically, the calculation equation are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}(y_i - \hat{y}_i)^2} \tag{11}$$

Here y_i and $\hat{y}_i$ represent the predicted value and real value at the i-th point at time t, respectively. Generally, lower MAE and RMSE values indicate better model prediction performance.

5.3.*Comparison and Result Analysis*

Table I show different imputation performence with different missing rate. MBGAN performs well in this task due to using the feature extraction module and attention mechanism.

Table 1. Imputation Performernce Under Different Missing Rates

	10% MISSING RATE		20% MISSING RATE		30% MISSING RATE	
MODEL	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
E^2GAN	25.83	26.90	25.87	27.89	26.84	28.54
BIGAN	24.83	26.55	24.94	27.85	24.84	27.54
MBGAN	10.83	4.55	11.31	16.83	12.61	17.47

6. CONCLUSION

Through the comparative analysis of the three GAN-based time series data imputation models of E^2GAN,BIGAN and MBGAN, it can be seen that the idea of confrontation adopted by GAN makes the process of data filling more intuitive. By further optimizing and integrating the input data characteristics and combining with the deformation of RNN, high-quality data can be generated. In order to further improve the recognition ability of the discriminator, it can also be optimized by RNN and other structures. In the future, how to improve the stability and speed of GAN model training can be further studied.We hope the generative adversarial network can be more practical and applied in time series data filling.

7. ACKNOWLEDGMENT

This research was supported by the Natural Science Foundation of Gansu Province (22JR5RA186). The authors would like to express their sincere thanks for this and would like to express their gratitude to the reviewers and editors for their comments.

References

[1] AMIRI M,JENSEN R.Missing data imputation using fuzzy-rough methods[J].Neurocomputing 2016,205,pp. 152–164.

[2] Gowtham Atluri ,Anuj Karpatne,and Vipin Kumar.2018.Spatio-temporal data mining:A survey of problems and methods.Compute. Surveys51, 2018,4, pp.1–41.
[3] Bertsimas,D.,Orfanoudaki,A.,Pawlowski,C.Imputation of clinical covariates in time series.Mach. Learn.2021,110 (1), pp, 185–248.
[4] Yoon,J.,Zame,W.R.,van der Schaar,M.Estimating Missing Data in Temporal Data Streams Using Multi-Directional Recurrent Neural Networks.IEEE Trans.Biomed.Eng.2019,66,pp,1477–1490.
[5] Song,S.,Sun,Y.,Zhang,A.,Chen,L.;Wang,J.Enriching Data Imputation under Similarity Rule Constraints.IEEE Trans.Knowl.Data Eng.2020,32,pp. 275–287.
[6] Miao,X.,Wu,Y.,Wang,J.,Gao,Y.,Mao,X.,Yin,J.Generative semi-supervised learning for multivariate time series imputation.In:Proceedings of the AAAI Conference on Artificial Intelligence.2021,vol.35,pp.8983–8991.
[7] W.Cao,D.Wang,J.Li,H.Zhou,L.Li,andY.Li.BRITS:bidirectional recurrent imputation for time series.In Advances in Neural Information Processing Systems 31:Annual Confer ence on Neural Information Processing Systems 2018,NeurIPS 2018,3-8 December 2018,Montr´eal,Canada,pp. 6776–6786.
[8] Luo,Y.,Zhang,Y.,Cai,X.,Yuan,X.E2GAN:End-to-End Generative Adversarial Network for Multivariate Time Series Imputation.In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence,IJCAI-19,Macao,China,10–16 August 2019,pp.3094–3100.
[9] Gupta M and Rahmatollah Beheshti.Time-series Imputation and Prediction with Bi-Directional Generative Adversarial Networks.2020, arxiv:https://arxiv.org/abs/2009.08900
[10] Gupta M , Phan T , Bunnell H T , et al. Concurrent imputation and prediction on EHR data using bi-directional GANs: Bi-GANs for EHR imputation and prediction[J]. ACM, 2021.
[11] NI Q, CAO X. MBGAN: An Improved Generative Adversarial Network with Multi-Head Self-Attention and Bidirectional RNN for Time Series Imputation[J]. Engineering Applications of Artificial Intelligence, 2022, 115, pp. 105232.
[12] Goodfellow,I.J.,Pouget-Abadie,J.,Mirza,M.,Xu,B.,Warde Farley,D.,Ozair,S.,Courville,A.,Bengio,Y. Generative adversarial nets.In:Proceedings of the 27th International Conference on Neural Information Processing Systems,Canada,Vol.2.Montreal,2014,pp. 2672–2680.