Transformer-based Spatial-Temporal Graph Attention Network for Traffic Flow Prediction

Fangzhou Yan, Qi Chen
School of AI and Advanced Computing
Xi'an Jiaotong Liverpool University
Suzhou, China
fangzhou.yan19@student.xjtlu.edu.cn
qi.chen02@xjtlu.edu.cn

***Abstract*—Traffic flow prediction, which plays an important role in intelligent traffic systems, has become a pressing problem to be addressed with the continuous development of smart cities. Currently, the fundamental obstacle lies in effectively modelling the complex spatial-temporal dependencies present in traffic flow data. Deep learning models such as Graph Neural Network based models and Transformer based models have shown promising results in this field. However, methods founded on a single model or framework have one significant limitation: Such methods cannot adequately represent the spatial and temporal features of traffic flow data, restricting the model's ability to learn the dynamics of urban transportation. In this paper, we propose a transformer-based spatial-temporal graph attention network model called TSTGAT for traffic flow prediction, which integrates Transformer and Graph Attention Network. Experiments on two real-world traffic datasets from the Caltrans Performance Measurement System (PeMS) demonstrate that the proposed TSTGAT model outperforms well-known baselines.**

***Index Terms*—Traffic flow prediction, transformer, graph neural network, deep learning**

I. Introduction

The intellectualization of cities has been the primary direction of development in numerous countries for a very long time. People are therefore committed to researching and developing intelligent transport systems (ITS) for efficient traffic management [1]. As an integral component of ITS, traffic flow prediction has been implemented to predict the future flow of traffic based on historical data observed by detectors. Predicting traffic flow can assist in achieving efficient traffic management, particularly on highways with high traffic volumes. By accurately predicting flow data in advance, traffic management departments can implement traffic control more rationally and enhance the operational efficiency of the highway network. Nevertheless, exploiting nonlinear and complex spatial-temporal dependent flow data to accurately predict traffic flow is a very difficult problem.

Time series models such as Recurrent Neural Network (RNN) and Long Short-term Memory(LSTM) have shown favourable performance in traffic flow prediction tasks since the data are temporal dependent. However, ignoring the spatial dependencies limits the prediction accuracy of the model. Graph Neural Network (GNN) based models are therefore used to capture spatial dependencies of the data. To achieve higher accuracy, people have attempted to combine LSTM, GNN and other techniques such as Transformer. Despite the fact that several studies have successfully integrated GNN and LSTM, it remains an open problem that how to better capture long-term dependency and spatial-temporal correlations.

In this paper, we propose a model utilizing Transformer and Graph Attention Network (GAT) to capture long-term dependency and spatial-temporal features and achieve promising results on two real-world traffic flow datasets. The rest of the paper is organised as follows. Section II begins with a summary of relevant works on traffic flow prediction. Then, in Section III, we introduce the architecture of our proposed model. Section IV demonstrates experimental results compared to other models. In Section V, we conclude the paper and discuss future research.

II. Related Work

To predict traffic flow, a large number of methods comprising various models have been applied. Methods could be divided into three categories: traditional statistical methods, classical machine learning methods and deep learning methods. A statistical method such as Autoregressive Integrated Moving Average model (ARIMA) [2] can capture traffic peaks and valleys through its obvious seasonality. As an example of machine learning methods, Support Vector Regression (SVR) [3] has contributed to the improvement of prediction accuracy by capturing the complexity of traffic flow data. Deep learning methods including STGCN [4], ASTGCN [5], and PDFormer [6], have made significant contributions to the advancement of traffic flow forecasting. STGCN extracts features from the spatial and temporal domains using graph convolutional networks. It can only be used to process data with relatively straightforward spatial and temporal relationships due to the fact that it is merely based on GNN. The attention-based design of ASTGCN compensates for the shortcomings

2833-8898/23/$31.00 ©2023 IEEE
DOI 10.1109/CyberC58899.2023.00031

of the STGCN. It is appropriate for processing data with complex spatial-temporal relationships. Moreover, PDFormer is a Transformer based model that employs the graph masking method to model local geographic domains and global semantic domains in spatial domains. With the emergence of Transformer models, it is necessary to explore the possibility of incorporating Transformer and GNN into traffic flow prediction models in order to enhance their precision.

III. Methodology

A traffic network could be represented as a graph $G = (V, E, A)$, where $V = \{v_1, v_2, \ldots, v_N\}$ representing N sensors, E is a set of edges reflecting connectivity between sensors and A is the adjacency matrix of the network G. Each node on the traffic network G records a set of features including total flow, average speed, and average occupancy at each time step. We use $X_t \in \mathbf{R}^N$ to represent the traffic flow at time t. The data observed by N sensors of historical H time steps could be denoted as $X = (X_{t_1}, X_{t_2}, \ldots, X_{t_H}) \in \mathbf{R}^{H \times N}$. Our purpose is to predict future P time steps for all traffic sensors, which is $Y = (\hat{X}_{t_{H+1}}, \hat{X}_{t_{H+2}}, \ldots, \hat{X}_{t_{H+P}}) \in \mathbf{R}^{P \times N}$.

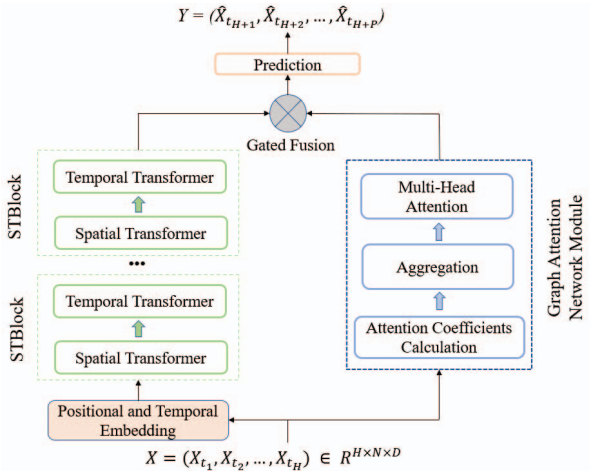

Fig. 1. The overall architecture of the proposed TSTGAT model.

Figure 1 illustrates the framework of our proposed TSTGAT model, which has an embedding module, a prediction layer and two main components: spatial-temporal blocks and GAT module. We use a 1×1 convolutional layer for embedding to expand the input dimensions to improve the expression ability of the model. Then, we extract spatial and temporal features of the data sequentially in spatial-temporal blocks. The correlation information between nodes are simultaneously captured by the GAT module. The results of STBlock and GAT are subsequently merged using a gated fusion mechanism for the final predicted results.

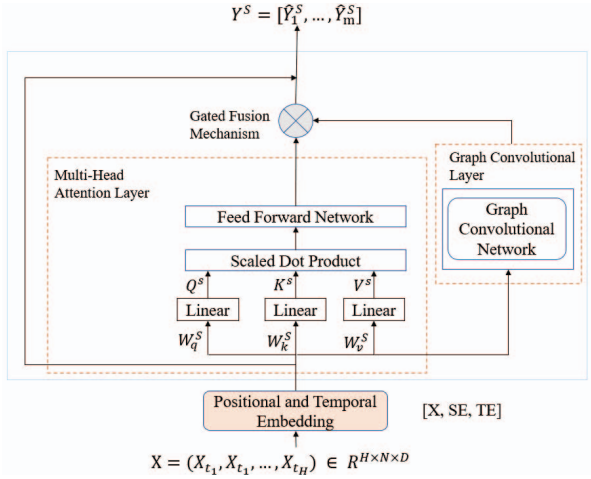

Fig. 2. Spatial Transformer in STBlock

A. Spatial and Temporal Transformer Block (STBlock)

As shown in Fig. 1, each STBlock incorporates one spatial and one temporal transformer module to jointly model the spatial and temporal dependencies of traffic networks for accurate prediction results. The input denoted as $X \in R^{H \times N \times D}$ is a 3-D tensor representing traffic flow information with time steps H, sensors N and features D. Normalization and residual connection are applied for stable training and model optimization. The spatial transformer module STN extracts spatial-temporal features Y^S through the input X^S and adjacency matrix A. The input X^T to the temporal transformer module TTN is generated by combining Y^S and X^S, and is utilized to extract temporal features. Stacking multiple STBlocks can increase feature expression ability, thereby better capturing spatial-temporal dependencies and conducting accurate prediction.

Figure 2 illustrates that STN consists of Graph Convolutional layer, Multi-Head Attention layer and gated fusion mechanism. Two trainable tensors spatial embedding $SE \in R^{N \times N}$ and temporal embedding $TE \in R^{H \times H}$ are initialised and concatenated with the input tensor in the spatial and temporal embedding layer [7]. Instead of modelling time and space separately, we employ spatial-temporal joint modelling. Through spatial-temporal joint modeling, spatial and temporal features can be fused together, enabling the model to better capture spatial-temporal features, thereby improving the predictive performance of the model. The Graph Convolutional layer uses GCN to learn the structural node features of graph G [8]. Since graph G is constructed based on the topology and fixed distances generated by physical connectivity between sensors, GCN can capture stationary spatial dependencies from the traffic network. Multiple sub GCNLayers are stacked to create the whole Graph Convolutional layer implementing the

the propagation and aggregation of node features. A GCNLayer $H^{(l+1)}$ can be expressed by:

$$H^{(l+1)} = \text{dropout}\left(\text{ReLU}\left(\widetilde{D}^{-1}\tilde{A}H^{(l)}W^{(l)}\right)\right) \quad (1)$$

where $\tilde{A} = A + I \in \mathbf{R}^{N\times N}$ a new adjacency matrix obtained by adding adjacency matrix A and identity matrix I. $\tilde{D}^{-1} \in \mathbf{R}^{N\times N}$ is a normalized diagonal matrix of D (degree matrix) aimed to normalize $\tilde{A}$. $H^{(l)} \in \mathbf{R}^{N\times D}$ is the hidden node features in the l-th layer; $H^{(0)}$ is the input X. $W^{(l)} \in R^{D\times h}$ is a trainable weight matrix. ReLU is used as the activation function and dropout is used for regularization.

As the GCN model only capture the static spatial-temporal dependencies while traffic network consists of dynamic spatial dependencies, we apply the Multi-Head Attention of Transformer [9] to capture dynamic spatial-temporal dependencies that evolve over time. The formulas could be defined as:

$$\text{Attention}(Q^S, K^S, V^S) = \text{softmax}\left(\frac{Q^S K^S}{\sqrt{d_k^S}}\right)V^S \quad (2)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \ldots, head_h\right)W^o \quad (3)$$

where $head_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$. Here, a Scaled Dot-Product Attention (2) and a Multi-Head Attention (3) are included. In formula (2), Q is the query vector matrix, K is the key vector matrix, V is the value vector matrix and d_k represents the vector dimension. In the formula (3), h is the number of heads and W^O is a linear transformation of the output vector matrix. Each head i maps the input vectors Q, K, and V onto the corresponding subspace through an independent linear transformation.

The architecture of temporal transformer module TTN is similar to STN excepting two components: 1) TTN merely conducts the temporal embedding TE, 2) TTN drops the GCN component. The reason is that the temporal transformer model is primarily used to model time series, and the influence of spatial features on time series is not significant.

B. GAT Module

The Graph Attention Network (GAT) is a typical graph neural network architecture proposed by Veličković et al [10]. Figure 1 illustrates that the GAT module consists of three parts which are Attention Coefficients Calculation, Aggregation and Multi-Head Attention. The following three Equations (4) - (6) correspond to the above three parts in order:

$$a_{ij} = \text{softmax}\left(\frac{e_{ij}}{\sum_{k\in\mathcal{N}_i} e_{ik}}\right) \quad (4)$$

$$h_i' = \sum_{j\in\mathcal{N}_i} a_{ij}\cdot \mathbf{W}\mathbf{h}_j \quad (5)$$

$$\mathbf{h}' = \text{Concat}(\text{head}_k(\mathbf{W}_k, \mathbf{h})) \quad \text{for } k \in [1, K] \quad (6)$$

In formula (4), $e_{ij} = \text{LeakyReLU}(\mathbf{a}^T \cdot [\mathbf{W}\mathbf{h}_i \parallel \mathbf{W}\mathbf{h}_j])$ is the raw unnormalized attention score between node i and node j computed using a shared weight vector a^T, $\mathbf{h}_i$ and $\mathbf{h}_j$ are the feature representations of node i and node j, respectively, $\mathbf{W}$ is a learnable weight matrix and $\parallel$ denotes concatenation. The attention coefficient between node i and node j is computed using the dot-product attention mechanism. It measures the relevance of the features of node j with respect to node i. In formula (5), h_i' is the aggregated feature representation of node i and $\mathbf{W}$ is the same learnable weight matrix used for computing attention coefficients. This part aggregates the neighbouring node representations by a weighted sum. In the formula (6), head_k represents the output of the k-th attention head, $\mathbf{W}_k$ is a learnable weight matrix specific to the k-th attention head and $\mathbf{h}'$ is the concatenated output of all attention heads. This part employs multiple attention heads to capture diverse patterns and relationships in the graph. For K attention heads, the outputs of all heads are concatenated along the last dimension.

IV. Experiments

A. Datasets

We validate our model on two highway traffic datasets which are PeMS04 and PeMS08 from California. The traffic flow data are collected by Caltrans Performance Measurement System (PeMS) [11] every 30 seconds and then further aggregated into 5 minutes. There are three kinds of traffic measurements including total flow, average speed and average occupancy. In this study, we use the total flow measurement from the past hour to predict the flow for the next hour.

B. Settings

All experiments are carried on a machine with the NVIDIA V100-32GB GPU and 72 GB memory. We implement TSTGAT with Ubuntu 20.04, PyTorch 1.11.0 and Python 3.8. The proposed model is trained with the mean squared error (MSE) loss using the Adam optimizer for 5 patience with a batch size of 32. The initial learning rate is set to 1e-4 and decays at a rate of 0.75 for every two epochs. The feature total flow has been selected for training and predicting in our experiments.

C. Evaluation

We compared TSTGAT with 4 baseline models: HA, LSTM, STGCN [4], ASTGCN [5].

- **HA:** Historical Average method leverages the average value of the last hour to predict the next value.

- **LSTM:** Long short-term memory model is a variant of RNN.
- **STGCN:** This is a spatial-temporal feature capture model based on graph convolution network.
- **ASTGCN:** The method is an attention-based spatial-temporal feature capture model, where the spatial feature is captured by attention mechanism through graph convolution using Chebyshev polynomial as the kernel and temporal feature is captured by regular convolution.

TABLE I
AVERAGE PERFORMANCE OF MODELS ON PEMS04 AND PEMS08

Models	PEMS04		PEMS08	
	RMSE	MAE	RMSE	MAE
HA	57.14	39.76	48.03	33.52
LSTM	51.52	34.80	43.27	28.98
STGCN	42.37	27.63	33.87	22.35
ASTGCN	33.01	20.91	31.61	18.64
TSTGAT	**32.57**	**20.79**	**29.08**	**17.96**

Table 1 shows a comprehensive comparison among various traffic flow prediction models. On the PEMS04 dataset, the TSTGAT model achieved remarkable results with an impressive RMSE of 32.57 and an MAE of 20.79. Similarly, on the slightly smaller-scale PEMS08 dataset, the TSTGAT model continued to excel, achieving an RMSE of 29.08 and an MAE of 17.96, further affirming TSTGAT's prowess in handling traffic data effectively. The TSTGAT model demonstrated the best performance in terms of RMSE and MAE, highlighting its exceptional accuracy and robust capability in the task of traffic flow prediction. These findings demonstrated the advantages of the novel Transformer and GAT integration method in TSTGAT for capturing spatial-temporal traffic patterns, predicting future traffic volume, and making it the preferred model for tackling traffic flow prediction challenges.

The traffic flow data is visualized in Figure 3. The figure shows the prediction results of TSTGAT, ASTGCN and the ground truth data for the first three days (864 time steps) from the same sensor. It can be seen from the figure that the prediction result of TSTGAT is closer to the ground truth compared with ASTGCN, which confirms that the proposed model is able to capture traffic flow patterns and provide better prediction results.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a model TSTGAT that provides a novel method for the integration of Transformer and Graph Attention Network (GAT) in traffic flow prediction to improve urban transportation management and decision-making. For future work, we would like to improve our model from 2 aspects: 1) External factors such as weather conditions, festivals and traffic accidents will be encoded into a data embedding layer to give more comprehensive prediction results. 2) Propagation delay will be processed in spatial and temporal self-attention layers as the occurrence of an accident takes time to impact traffic conditions in adjacent areas. By addressing these issues, we aspire to create a comprehensive traffic flow prediction model that demonstrates robustness in handling complex and large-scale real-world scenarios.

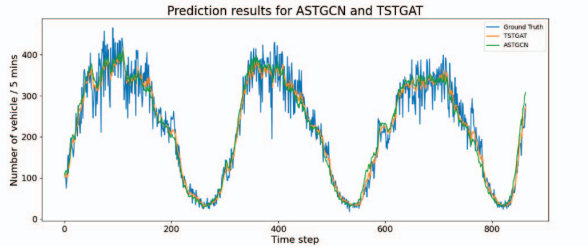

Fig. 3. Traffic flow data predicted by different models

ACKNOWLEDGMENT

This research is supported by the Jiangsu Science and Technology Programme (BK20221260) and the Research Development Fund (RDF-22-01-132) at XJTLU.

REFERENCES

[1] ChuanTao Yin, Zhang Xiong, Hui Chen, JingYuan Wang, Daven Cooper, and Bertrand David. A literature survey on smart cities. *Sci. China Inf. Sci.*, 58(10):1–18, 2015.

[2] Mohammed S Ahmed and Allen R Cook. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. Number 722. 1979.

[3] Shangyu Sun, Huayi Wu, and Longgang Xiang. City-wide traffic flow forecasting using a deep convolutional neural network. *Sensors*, 20(2):421, 2020.

[4] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[5] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 922–929, 2019.

[6] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction. *arXiv preprint arXiv:2301.07945*, 2023.

[7] Dachuan Liu, Jin Wang, Shuo Shang, and Peng Han. Msdr: Multi-step dependency relation networks for spatial temporal forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1042–1050, 2022.

[8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[11] Caltrans. Performance measurement system (pems), 2023. [Online] http://pems.dot.ca.gov.