Received: September 1, 2023. Revised: October 7, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Traffic Flow Prediction and Identifying Critical Node Networks Using Attention-based Spectral Temporal Graph Neural Network

Sreenath Marimakalapalli Venkatarama Reddy[1]* **Annapurna Dammur[1]**
Anand Narasimhamurthy[2]

[1]*Department of Computer Science and Engineering, PESIT Bangalore South Campus, Bangalore 560100, India*
[2]*Department of Computer Science and Engineering, INSOFE, Bengaluru 560102, India*
* Corresponding author's Email: sreenathmv@pes.edu

Abstract: The accurate and dependable prediction of traffic is essential for the stable and safe implementation system of intelligent transportation. It is difficult to examine both inter-series and intra-series temporal correlations at the same time. However, several recent research in the time domain have attempted to capture both correlations but numerous analyses only capture the correlations of temporal and rely on pre-determined priors as inter-series interactions. In this paper, an attention-based spectral temporal graph neural network (AStemGNN) is proposed for traffic flow prediction and identifying the critical node/link graph network. In the spectrum domain, both the correlations of inter-series and temporal dependency are captured by AStemGNN. It incorporates graph fourier transforms (GFT), which describe the correlations of inter-series, and discrete fourier transform (DFT) which represent temporal dependencies. Following the execution of GFT and DFT, the spectrum representations retain patterns and it was predicted efficiently by the modules of convolution and sequential learning. Furthermore, an AStemGNN automatically identifies the correlations of inter-series data without employing the help of pre-defined priors. The efficacy of the AStemGNN method is demonstrated by employing the PEMS04 and PEMS08 datasets. The existing methods such as attention-based spatial-temporal graph neural networks (ASTGNN), static and dynamic spatial correlation neural networks (SDSCNN), dynamic graph convolutional recurrent imputation networks (DGCRIN), and graph and attentive multi-path convolutional network (GAMCN) are used to justify the efficacy of the AStemGNN. When compared with the existing methods such as ASTGNN, SDSCNN, DGCRIN, and GAMCN, the AStemGNN achieves 0.65%, 1.12% mean absolute error (MAE), 0.78%, 1.05% root mean squared errors (RMSE), and 1.57%, 2.19% mean absolute percentage error (MAPE) in both PEMS04, PEMS08 datasets respectively.

Keywords: Graph neural network, Inter-series correlations, Intra-series temporal correlations, Link/node identification, Traffic flow prediction.

1. Introduction

Traffic flow prediction is a technique that examines current traffic circumstances on the urban road network, mining traffic patterns, and forecasts future traffic conditions on the road network such as density, flow, and speed [1]. Traffic flow forecast is considered to be a vital part of intelligent transportation systems (ITS), offering significant benefits in the reduction of traffic congestion, enhancing the efficiency of traffic control, decreasing traffic accidents, and timely resource deployment [2]. Real-world applications like the recognition of skeleton-based action and traffic prediction frequently use spatial-temporal data [3]. Accurate spatiotemporal traffic flow forecasting is required to prevent public transport congestion. This can enable decision-making for the management of traffic, including temporary traffic control, and traffic signal modification [4]. Due to the spatial dependence between adjacent road segments, the prediction of spatial can be done by estimating traffic flows from nearby roadways [5]. Spatial-temporal graph modeling has gained increased attention as graph neural networks develop to achieve more accurate traffic predictions in long-term and complex spatial situations [6]. Numerous different types of sensor devices have been used on the transportation network due to the advancement of sensor technology [7]. These sensors generate a lot of geographic-based traffic data, which offers adequate traffic forecasting [8]. Numerous different types of sensor devices have been used on the transportation network due to the advancement of sensor technology.

A key element of crowd control and transit management in urban rail transit is the short-term forecasting flow of passengers [9]. Using short-term passenger flow forecasting, transit operators can manage passenger flow to prevent congestion and deliver real-time traffic information to passengers to assist them in making intelligent scheduling decisions [10]. Mobility patterns and travel demand are frequently the most important inputs for planning and operating transportation infrastructure/services such as rail lines, dedicated bus lanes, and scheduling, which further determine the public transit service's attraction, effectiveness, and dependability [11]. Traffic flow spreads during several traffic nodes, and downstream node flow is closely related to upstream node flow. There are typically both long-term and short-term neighboring periodic temporal dependencies for the temporal feature extraction [12]. Although network traffic shows enormous variation over time, it is still possible to forecast traffic using its periodic patterns [13]. To predict traffic, time series forecasting models that emphasize using temporal characteristics, such as autoregressive integrated moving average (ARIMA) and recurrent neural network (RNN) are frequently utilized [14]. Some research employs convolutional neural networks (CNN) to extract spatial information and combine it with long-short term memory (LSTM) to increase the prediction accuracy to assess the spatial dependence of traffic flows [15]. However, several attempts have recently tried to capture both correlations but only capture correlations of temporal in the time domain and rely on inter-series interactions with pre-defined priors. To overcome these issues, following are the primary contribution of this paper is summarized below:

- In the spectrum domain, attention-based spectral temporal graph neural network (AStemGNN) effectively represents inter-series and intra-series correlations. The advantages of discrete fourier transform (DFT), graph fourier transform (GFT), and deep neural networks are combined and implemented.
- For various time series, AStemGNN provides data-driven dependency graph generation. As a result, the model can be used for any time series that lacks predefined topologies.
- Graph neural network (GNN) has been established for the effective identification of critical links/nodes in the network of large complexes and is used to learn link/node criticality scores in a variety of applications, including urban, social, and biological networks.

The remainder of the paper is represented as follows. Section 2 describes the literature survey. Section 3 discusses the problem statement. Section 4 describes the objectives. Section 5 discusses the proposed methodology. Section 6 describes the results. Section 7 discusses the conclusion.

2. Literature survey

Guo [16] implemented an attention-based spatial-temporal graph neural network (ASTGNN) for predicting the traffic flow in both the dimensions of temporal and spatial, including the heterogeneity and periodicity of spatial data. To include a time series local context, a trend-aware muti-head attention method was established specifically for prediction time series tasks. The implemented method achieves the best-predicting performance by accurately modeling the traffic data dynamics. However, ASTGNN heavily depends on previous traffic data, this performance suffers in limited or noisy environments and does not generalize well to new traffic patterns.

Dai [17] presented static and dynamic spatial correlation neural networks (SDSCNN) for the prediction of traffic flow and a network of graph attention was employed to generate the modules of static and dynamic using the correlation between traffic and distance data from the road network. Using the multi-head self-attention method, the temporal correlation and traffic flow periodicity were then recorded, and several spatial-temporal layers were combined for prediction. METR-LA, PEMS04, and PEMS08 were the three datasets that showed the implemented method achieves the performance of good prediction. However, the model's prediction performance was impacted by the relatively homogenous characteristics of the input data, which include car accidents, heterogeneous weather, and unexpected situations.

Tang [18] introduced a spatial-temporal graph attention-based dynamic graph convolutional network (GAGCN) for the flow of traffic prediction. GAGCN uses graph attention networks to automatically extract spatial connections among hidden nodes in the traffic data feature that was dynamically changed over time. To acquire the spatial properties of the road network, the graph convolutional network is then modified based on the connections of spatial. GAGCN has greater versatility and accuracy and also improves prediction performance. However, the capacity of GAGCN to generalize diverse traffic situations or geographic locations was limited.

Ke [19] implemented an automated spatio-temporal graph prediction (AutoSTG) to learn the adjacency matrices of spatial graph convolution layers and kernels of temporal convolutional layers from metagraph information. Then meta-learning was employed to produce the SC diffusion matrices and the layers of TC kernels from the acquired properties. The implemented method maximizes the search space and includes new graphs that enhance and stabilize the performance of the model. However, the search space was constrained or not sufficiently representative of the various network designs, and the implemented method produced biased or inaccurate outcomes.

Kong [20] presented a dynamic graph convolutional recurrent imputation network (DGCRIN) to compute missing traffic data. To achieve fine-grained modeling of the dynamic spatiotemporal dependencies of the road network, DGCRIN employs a generator of the graph and dynamic graph convolutional gated recurrent unit (DGCGRU). This method enables predictions more reliable and precise, even when there are missing points of data. However, traffic data typically establish substantial dynamic correlations in the spatiotemporal dimension, for a static graph structure was not properly provided.

Qi [21] introduced a graph and attentive multi-path convolutional network (GAMCN) to forecast short-term traffic situations and concentrate on the temporal and spatial variables that affect traffic conditions. To acquire the correlations of spatial and temporal traffic situations, the GAMCN system combines the component of enhanced GCN with a precise multi-path CNN component. The outcomes indicate that the implemented method performs well in terms of prediction efficiency and errors. However, for effective training, the GAMCN method needs a large amount of historical traffic data including both temporal and spatial information.

Zhang [22] implemented an evolution temporal graph convolutional network (ETGCN) method to create a spatial correlation and then, to predict the speed of traffic on a road network graph, the spatial-temporal dependence and their dynamic changes were simultaneously learned. To fuse various network adjacency matrices, a similarity-based attention technique was offered, and then a gated recurrent unit was coupled with GCN to simultaneously record the correlations of spatial-temporal. By using the SZ-taxi dataset, the technique generates better prediction outcomes. However, scaling ETGCN to address greater or more intricate spatial and temporal datasets was difficult.

There are some limitations with the existing methods that are mentioned above such as the DGCRIN's traffic data typically establish substantial dynamic correlations in the spatiotemporal dimension, for which a static graph structure was not properly provided. The GAMCN method needs a large amount of historical traffic data including both temporal and spatial information for training.

3. Problem statement

The problem found with the general issues in traffic flow prediction and critical node/link identification is discussed below:

- The challenge is to identify essential nodes or linkages in a road network that have a major effect on the overall flow of traffic and congestion.
- However, computation cost and training time were increased due to the optimization process use of large-scale graphs and tuning of hyperparameters.
- The issue of traffic flow forecasts is to accurately predict the flow of traffic in a given area at a specific period in the future.

4. Objectives

Traffic flow prediction and critical node/link identification are effectively carried out by utilizing deep learning techniques.

- The main objective is to identify critical nodes or links that are essential for optimizing the management of traffic measures such as lane additions, adjustments of signal timing, and infrastructure enhancement.
- To develop scalable and effective methods for reducing the computational resources-related constraints on the prediction of traffic flow and critical node identification.
- The goal is to reduce forecasting errors and increase the accuracy of the prediction.

Received: September 1, 2023. Revised: October 7, 2023.

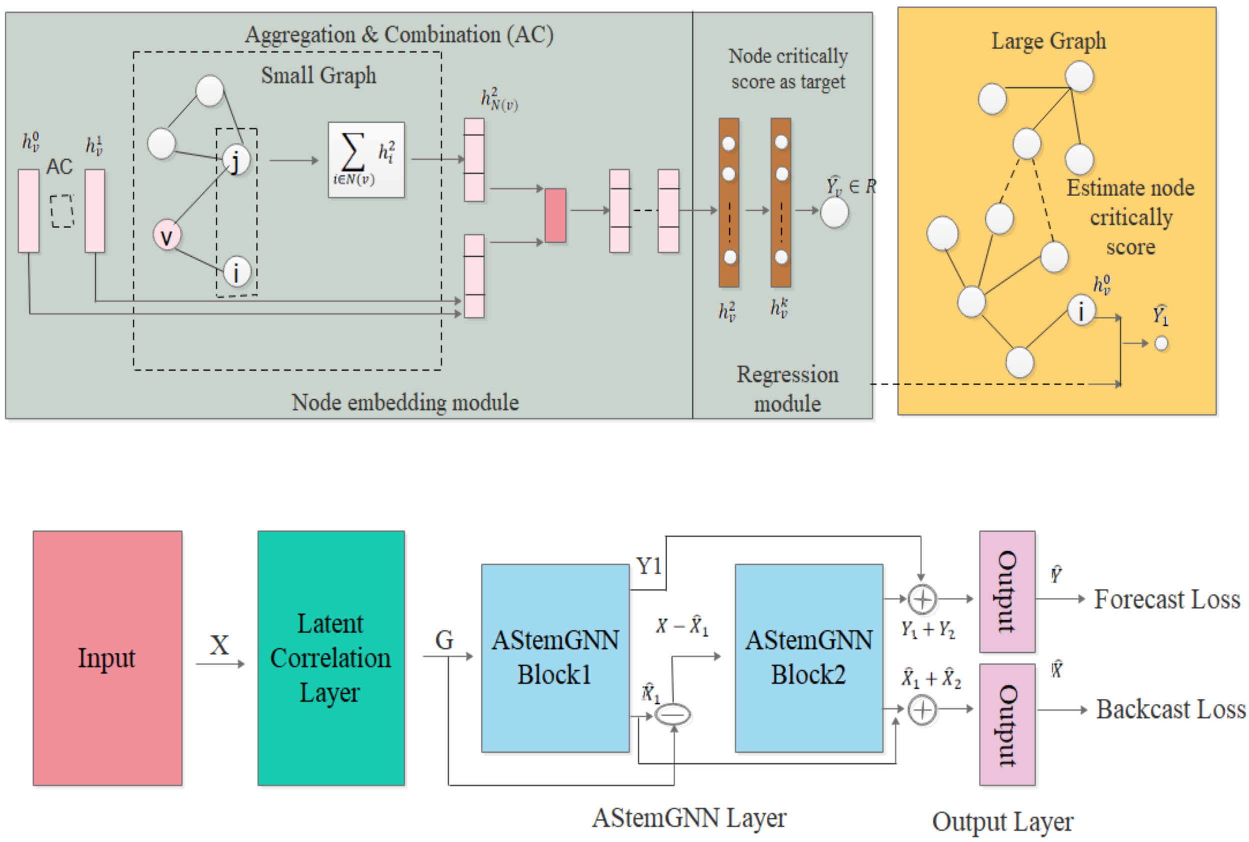

Figure. 1 Block Diagram of the proposed method

5. Proposed methodology

For traffic flow prediction and identifying the critical node/link graph network an AStemGNN is proposed. The overview of the proposed method is indicated in Fig. 1.

5.1 Input and latent correlation layer

First, the layer of latent correlation is supplied a traffic flow time-series input called X where the weight matrix W and the structure of the graph can be determined automatically from the data. The AStemGNN layer, which consists of two residual AStemGNN blocks, is then input by the graph $G = (X, W)$. When modeling multivariate time series, a GNN-based technique requires the use of a graph structure. It is built using human knowledge like traffic road network prediction, however there are occasions when there is no prior predefined structure of the graph. The self-attention technique is used to detect latent correlations between several time series to serve general cases. The model shows correlations that are relevant in a data-driven manner. The income $X \in R^{NXT}$ is received by the layer of GRU and the hidden state associated with every timestamp t is determined. The W weight matrix is then determined by the process of self-attention representing the entire time series using the hidden state R is shown in Eq. (1)

$$Q = RW^Q, K = RW^K, W = Softmax(\frac{QK^T}{\sqrt{d}}) \quad (1)$$

Where $d - Q$ and K hidden dimension size
Q, K – Query and Key representation
$W \in R^{NXN}$- graph G's adjacency weight matrix
$O(N^2 d)$ - time complexity of self-attention

5.2 Attention-based spectral temporal graph neural network (AStemGNN) block

AStemGNN block was created specifically to model both dependencies of structural and temporal within traffic flow time series in the domain spectrum. A Spe-Seq cell is used to create an AStemGNN block by combining it with a module of spectral graph convolution. In the spectral domain, due to its exceptional capacity to acquire latent structures of several time series, spectrum graph convolution is frequently employed in a series of time prediction tasks. The Graph G is first converted into a spectral matrix representation using a GFT, which leads to the univariate time series for each node to become linearly independent. Then, every univariate series of time is changed into the frequency domain employing

 DOI: 10.22266/ijies2023.1231.78

a DFT operator. The representation is supplied through GLU sub-layers and 1D convolution in the frequency domain to collect pattern features before being translated back to the domain time using inverse DFT. In the end, inverse GFT is performed on the representation of the spectral matrix and applied convolution of the graph. The remaining elements of the Spectral Graph Convolution are then used to process the Spe-Seq cell output. To calculate the outcomes after the convolution of the graph with kernel Θ_j, graph fourier transform (GFT) and spe-seq cell are applied to every individual channel of input data X_i. Then, to acquire the j^{th} channel Z_j of the output, Inverse GFT is included in the sum as follows in Eq. (2)

$$Z_j = GF^{-1}(\sum g\Theta_{ij}(\Lambda_i)S(GF(X_i))) \tag{2}$$

Where GF, S, GF^{-1} – GFT, Spe-Seq Cell, IGFT

Θ_{ij} - graph convolution kernel for the channel of input i and output j

Λ_i - normalized Laplacian matrix of eigenvalue

Based on Z, basis expansion coefficients θ are produced by a fully connected layer using learnable parameters to determine V basis vectors. The output can then be determined using several various bases $Y = V\theta$. In the AStemGNN block, this module has two branches: the branch of the forecast, which predicts future values, and the branch of backcast represented by B, which reconstructs historical values. The backcasting branch aids in controlling the block's functional space while representing the data of the time series. To construct more complex models, residual connections are used to stack different AStemGNN blocks. Two AStemGNN blocks are employed in this instance. Next block attempts to estimate difference among the reconstructed values of first block and ground truth. To provide predictions, both blocks' outputs are combined and transmitted into fully connected layers and GLU.

5.2.1. Spectral sequential cell (spe-seq cell)

The spe-seq cell S seeks to learn the feature representations on each unique series of time after decomposing GFT on the basis of frequency. GLU, DFT, F, Inverse DFT, F^{-1}, and 1D convolution are its four components where IDFT and DFT transform time-series information among the frequency and temporal domains, while in the domain of frequency represents GLU learn feature representation and 1D convolution. The real part $\widehat{X_u^r}$ and imaginary part $\widehat{X_u^i}$ of the DFT outcome are specifically managed by identical operators with various parallel conditions. The operations can be written as shown in eq. (3).

$$\begin{aligned} M^*(\widehat{X_u^*}) &= GLU(\theta_T^*(\widehat{X_u^*}), \theta_T^*(\widehat{X_u^*})) \\ &= \theta_T^*(\widehat{X_u^*}) \odot \sigma^*(\theta_T^*(\widehat{X_u^*})), * \epsilon \{r, i\} \end{aligned} \tag{3}$$

Where θ_T^* - convolution kernel

$\odot$ - Hadamard product and the sigmoid gate of nonlinear

σ^* - calculates the degree to which the data in the present input is closely associated. Finally, $M^r(\widehat{x_u^r}) + iM^i(\widehat{x_u^i})$ and IDFT is used for the outcome.

5.2.2. Spectral graph convolution

Three phases make up the spectral graph convolution: first, the input series of time is transferred to the spectrum domain by GFT. Secondly, the spectrum characterization is processed by the operator of the graph convolution by kernels. Finally, to get the outcome, the spectral representation undergoes the IGFT. One of the essential operators for spectral graph convolution is GFT. The laplacian eigenvectors of the normalized graph are used to build the bases of the orthonormal space where the input graph is estimated. The Laplacian of the normalized graph is calculated in Eq. (4).

$$L = I_N - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{4}$$

Where $I_N \epsilon R^{NXN}$ – identity matrix

D - degree matrix in a diagonal

Then, using the Laplacian matrix's eigenvalue decomposition obtained L in Eq. (5)

$$L = U \Lambda U^T \tag{5}$$

Where $U \epsilon R^{NXN}$ – eigenvalue matrix

Λ - eigenvalue diagonal matrix

The GFT and IGFT operators are described as $GF(X) = U^T X = \hat{X}$ and $GF^{-1}(\hat{X}) = U\hat{X}$. The $g\Theta(\Lambda)$ function of the eigenvalue matrix with parameter provides the graph convolution operator. $0(N^3)$ is the general temporal complexity.

5.3 Graph neural network (GNN)

A kind of artificial neural network called graph neural network (GNN) is created to identify patterns in data that are represented graphically. It is one of the initial works that effectively convert the convolutional operations from Euclidean to graph space. The models may operate directly on graphs and their topological information. Their operating concept is similar to that of convolutional neural networks (CNN). Classification of nodes, graphs, and link prediction, etc. are common training tasks for graph data. The embedding vectors of the learning node are often followed by layers of feedforward for regression or classification tasks in a GNN. Scalability problems arise since the proposed learning process is dependent on the graph size. Since subgraphs are used to learn node embeddings, and graph size has no impact on the way the training is carried out. This framework can also be used to infer information about previously unknown or new nodes in the network that belong to the same family. The typical method for learning this embedding vector is a mechanism of message-passing, in which data node features are gathered from a node's neighbors and merged with that node's features to create a vector of new features. To create the final embedding for each graph node, this process is repeated. Rather than learning the embedding vectors, the GraphSAGE algorithm learns the function of mapping. As a result, given its feature and neighborhood, it can induce the embedding of a new node or node unseen during training. GNN covers a range of learning challenges in a variety of fields, including natural language processing, biochemistry, and computer vision. The proper node embeddings and criticality scores are learned using computationally feasible graphs and a GNN, which speeds up the criticality score learning process. The GNN model learns node embeddings from a collection of node features, which are then employed to compute criticality scores.

6. Experimental setup and results

To assess the effectiveness of the AStemGNN, this paper employs the Operating system running on Windows (CPU: Intel Core i7-8700 @ 3.20GHz, GPU: NVIDIA GeForce GTX 1070Ti). The hyper-parameters are tuned using a grid search for AStemGNN on the validation data. Finally, kernel size for 1D convolution is set to 3 and the channel size for every layer of graph convolution is set to 64. After that, the optimizer of RMSprop is used, and 50 training epochs are selected.

6.1 PEMS04, PEMS08 datasets

Datasets are related to the flow of traffic on California's highways and are gathered in real-time, every 30 seconds, by the caltrans performance measurement system (PeMS). In a 5-minute interval, the raw traffic flow data is collected. The datasets include geographic data regarding the sensor stations. PEMS04 and PEMS08 are the two datasets generated for traffic flow prediction. This dataset is well-known for providing a benchmark in traffic prediction. In PEMS08 dataset, there are 1979 sensors on 8 roads, and 170 of them were chosen for prediction.

6.2 Evaluation metrics

Let T be the total number of timestamps, $\widehat{X_t}$ and X_t be the forecast and actual values at timestamp t. The experiments evaluation metrics can be calculated by following Eqs. (6), (7), and (8)

- Mean Absolute Error (MAE) – To assess a model's accuracy, Mean Absolute Error is utilized. Absolute numbers or positive numbers are used to determine the error.

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left|X_t - \hat{X}_t\right| \tag{6}$$

- Root Mean Squared Errors (RMSE) – It is obtained by square rooting the MSE.

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left|X_t - \hat{X}_t\right|} \tag{7}$$

- Mean Absolute Percentage Error (MAPE) – It is employed to determine a model's predicting accuracy.

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{X_t-\hat{X}_t}{X_t}\right| \, X\ 100\% \tag{8}$$

6.3 Experimental results

The performance of the ASTEMGNN is analyzed with different classifiers such as RNN, LSTM and GNN for different k fold sizes. The k-fold sizes considered for evaluating the ASTEMGNN are 3, 5 and 10. Further, the performances are analyzed for all features. The analysis of ASTEMGNN with different classifiers for all features is shown in Table 1 and Fig. 2. Similarly, the analysis of ASTEMGNN with different classifiers for selected features. Further, the graphical illustration of classification performances for different classifiers using all features and selected features with 10-folds are shown in Fig. 2 respectively. From the tables, it is found that the STEMGNN provides better performances than the RNN, LSTM, STEMGNN and GNN. For example, the accuracy of STEMGNN with the selected feature for 10-fold is 99.00 % whereas RNN obtains 74.92 %, RNN obtains 78.29%, GNN obtains 81.82 % and GNN obtains 81.74 %. The performances of STEMGNN are improved for the following reasons:

Received: September 1, 2023. Revised: October 7, 2023.

Table 1. The analysis of ASTEMGNN with different classifiers

Cross-folds	Measures	RNN	LSTM	GNN	STEMGNN	ASTEMGNN
3-folds	MAE	0.333	0.304	0.282	0.255	0.213
	RMSE	0.57	0.546	0.521	0.492	0.456
	MAPE	2.03	2.06	2.078	2.104	2.083
5-folds	MAE	0.189	0.162	0.143	0.117	0.066
	RMSE	0.433	0.408	0.379	0.362	0.316
	MAPE	2.026	2.05	2.078	2.102	2.089
10-folds	MAE	0.456	0.437	0.403	0.386	0.037
	RMSE	0.685	0.653	0.638	0.603	0.062
	MAPE	2.029	2.047	2.083	2.106	0.114

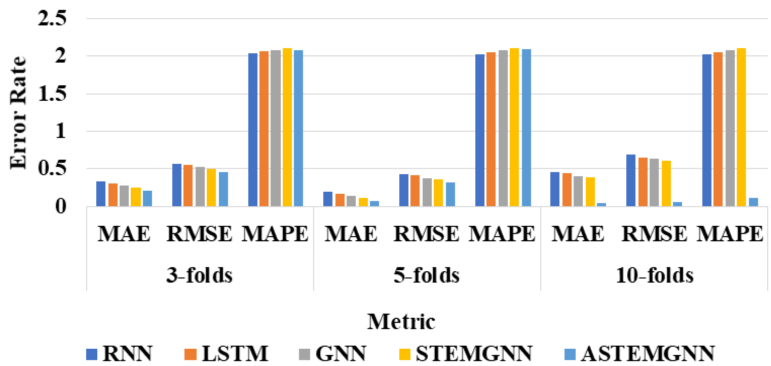

Figure. 2 The analysis of ASTEMGNN with different classifiers

Table 2. Comparative analysis with existing methods

Author	Methods	Datasets	MAE (%)	RMSE (%)	MAPE (%)
Guo [16]	ASTGNN	PEMS04	3.9	2.5	7.1
		PEMS08	4.5	2.12	8.4
Dai [17]	SDSCNN	PEMS04	2.86	0.91	2.59
		PEMS08	3.15	1.1	3.78
Kong [20]	DGCRIN	PEMS04	1.466	3.062	3.38
		PEMS08	1.114	2.467	2.28
Qi [21]	GAMCN	PEMS04	0.96	1.80	1.71
		PEMS08	1.22	2.65	2.41
Proposed Method	AStemGNN	PEMS04	0.65	0.78	1.57
		PEMS08	1.12	1.05	2.19

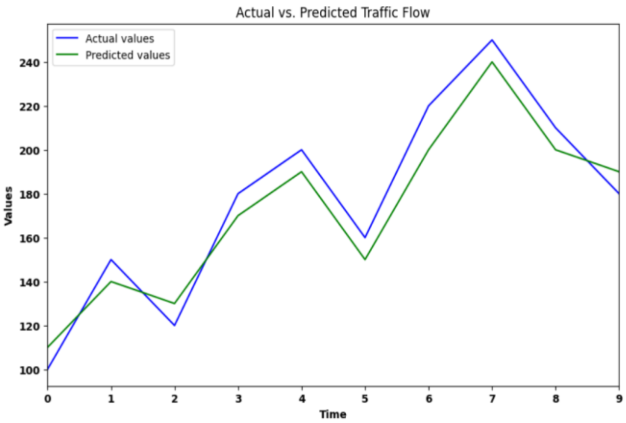

Figure. 3 Actual vs Predicted traffic flow

1) Utilization of Adam optimizer is used to optimize the target function that helps to minimize the error and 2) HLF is used to balance the MSE and MAE.

Fig. 3 shows the Actual vs Predicted traffic flow. The actual value is the value acquired from measurement or observation of the relevant data. It is also known as the observed value. The expected value is the predicted value of the variable based on the regression analysis. The most frequent method for calculating model error is linear regression by employing MSE. The difference between the actual and predicted values is used to calculate an error rate, and the goal is to minimize this difference.

6.4 Comparative analysis

The comparative analysis includes datasets, methods, MAE, RMSE, and MAPE. Table 2 shows the comparative analysis with the existing methods.

The existing methods ASTGNN [16] has a 3.9%, 4.5% MAE, 2.5%, 2.12% RMSE, and 7.1%, 8.4%

MAPE in both PEMS04, PEMS08 datasets. SDSCNN [17] has a 2.86%,3.15% MAE, 0.91%, 1.1% RMSE, and 2.59%, 3.78% MAPE in both PEMS04, PEMS08 datasets. DGCRIN [20] has a 1.466%, 1.114% MAE, 3.062%, 2.467% RMSE, and 3.38%, 2.28% MAPE in both PEMS04, PEMS08 datasets. GAMCN [21] has a 0.96%, 1.80% MAE, 1.80%, 2.65% RMSE, and 1.71%, 2.41% MAPE in both PEMS04, PEMS08 datasets. when compared with the existing methods, AStemGNN achieves 0.65%, 1.12% MAE, 0.78%, 1.05% RMSE, and 1.57%, 2.19% MAPE in both PEMS04, PEMS08 datasets.

6.5 Discussion

This section provides a discussion about the AStemGNN method and compares those results with existing methods in comparative analysis section 6.4. The major goal of this study is to predict the traffic flow and to identify the critical node/link graph network. In the spectrum domain, both the correlations of inter-series and temporal dependency are captured by AStemGNN. The Graph G is first converted into a spectral matrix representation using a GFT, which leads to the univariate time series for each node to become linearly independent. Then, each univariate series of time is converted into the frequency domain employing a DFT operator. Finally, inverse GFT is performed on the representation of the spectral matrix and applied convolution of the graph. A Spectral Sequential (Spe-Seq) cell is used to create an AStemGNN bock by combining it with a module of spectral graph convolution. In the spectral domain, due to its exceptional capacity to acquire latent structures of several time series, spectrum graph convolution is frequently employed in a series of time prediction tasks. The efficacy of the AStemGNN method is demonstrated by employing the PEMS04, PEMS08 datasets. When compared with existing methods ASTGNN [16], SDSCNN [17], DGCRIN [20], GAMCN [21], the AStemGNN achieves 0.65%, 1.12% MAE, 0.78%, 1.05% RMSE, and 1.57%, 2.19% MAPE in both PEMS04, PEMS08 datasets.

7. Conclusion

In this paper, the AStemGNN method is proposed to predict the traffic flow and to identify the critical node/link graph network. The advantages of DFT, GFT, and deep neural networks are combined and implemented. Furthermore, an AStemGNN automatically identifies the correlations of inter-series data without employing the help of pre-defined priors. The GNN has been established for the effective identification of critical links/nodes in the network of large complexes and is used to learn link/node criticality scores in a variety of applications, including urban, social, and biological networks. The efficacy of the proposed method is demonstrated by employing datasets of PEMS04 and PEMS08. The AStemGNN performance is simulated and compared with the existing methods such as ASTGNN, SDSCNN, DGCRIN, GAMCN. AStemGNN achieves 0.65%, 1.12% MAE, 0.78%, 1.05% RMSE, and 1.57%, 2.19% MAPE in both PEMS04, PEMS08 datasets. In the future, the approximation technique will be examined to decrease AStemGNN time complexity because directly using the decomposition of eigenvalue is prohibitively expensive for very large high-dimensional time-series graphs.

Notation Table

Symbol	Description
d	Q and K hidden dimension size
Q, K	Query and Key representation
$W \in R^{NXN}$	Graph G's adjacency weight matrix
$O(N^2 d)$	The time complexity of self-attention
GF, S, GF^{-1}	GFT, Spe-Seq Cell, IGFT
Θ_{ij}	Graph convolution kernel for the channel of input i and output j
Λ_i	Normalized Laplacian matrix of eigenvalue
$\widehat{X_u^r}$	a real part of GFT
$\widehat{X_u^i}$	the imaginary part of GFT
θ_T^*	convolution kernel
$\odot$	Hadamard product and the sigmoid gate of nonlinear
σ^*	calculates the degree to which the data in the present input is closely associated.
$I_N \epsilon R^{NXN}$	identity matrix
D	degree matrix in a diagonal
$U \epsilon R^{NXN}$	eigenvalue matrix
Λ	eigenvalue diagonal matrix
$0(N^3)$	general temporal complexity
t	timestamp
V	Basic vector
θ	expansion coefficients

Conflicts of interest

The authors declare that they have no conflict of interest.

Author contributions

For this research work all authors' have equally contributed in Conceptualization, methodology, validation, resources, writing—original draft preparation, writing—review and editing.

Received: September 1, 2023. Revised: October 7, 2023.

References

[1] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 6, pp. 3337-3348, 2021.

[2] S. Fang, V. Prinet, J. Chang, M. Werman, C. Zhang, S. Xiang, and C. Pan, "MS-Net: Multi-Source Spatio-Temporal Network for Traffic Flow Prediction", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 7, pp. 7142-7155, 2022.

[3] Z. Tan, Y. Zhu, and B. Liu, "Learning spatial-temporal feature with graph product", *Signal Processing*, Vol. 210, p. 109062, 2023.

[4] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, "A graph-based temporal attention framework for multi-sensor traffic flow forecasting", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 7, pp. 7743-7758, 2022.

[5] Z. Yi, X. C. Liu, N. Markovic, and J. Phillips, "Inferencing hourly traffic volume using data-driven machine learning and graph theory", *Computers, Environment and Urban Systems*, Vol. 85, p. 101548, 2021.

[6] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, "STGAT: Spatial-Temporal Graph Attention Networks for Traffic Flow Forecasting", *IEEE Access*, Vol. 8, pp. 134363-134372, 2020.

[7] J. Ye, S. Xue, and A. Jiang, "Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction", *Digital Communications and Networks*, Vol. 8, No. 3, pp. 343-350, 2022.

[8] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic Graph Convolution Network for Traffic Forecasting Based on Latent Network of Laplace Matrix Estimation", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 2, pp. 1009-1018, 2022.

[9] Y. He, L. Li, X. Zhu, and K. L. Tsui, "Multi-Graph Convolutional-Recurrent Neural Network (MGC-RNN) for Short-Term Forecasting of Transit Passenger Flow", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 10, pp. 18155-18174, 2022.

[10] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No. 11, pp. 7004-7014, 2021.

[11] C. Li, L. Bai, W. Liu, L. Yao, and S. T. Waller, "Graph Neural Network for Robust Public Transit Demand Prediction", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 5, pp. 4086-4098, 2022.

[12] Z. Xu, Y. Kang, Y. Cao, and Z. Li, "Spatiotemporal Graph Convolution Multifusion Network for Urban Vehicle Emission Prediction", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 32, No. 8, pp. 3342-3354, 2021.

[13] S. Fang, X. Pan, S. Xiang, and C. Pan, "Meta-MSNet: Meta-Learning Based Multi-Source Data Fusion for Traffic Flow Prediction", *IEEE Signal Processing Letters*, Vol. 28, pp. 6-10, 2021.

[14] K. He, X. Chen, Q. Wu, S. Yu, and Z. Zhou, "Graph Attention Spatial-Temporal Network With Collaborative Global-Local Learning for Citywide Mobile Traffic Prediction", *IEEE Transactions on Mobile Computing*, Vol. 21, No. 4, pp. 1244-1256, 2022.

[15] J. Zhu, X. Han, H. Deng, C. Tao, L. Zhao, P. Wang, T. Lin, and H. Li, "KST-GCN: A Knowledge-Driven Spatial-Temporal Graph Convolutional Network for Traffic Forecasting", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, No. 9, pp. 15055-15065, 2022.

[16] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 34, No. 11, pp. 5415-5428, 2022.

[17] J. Dai, J. Huang, Q. Shen, Q. Shi, S. Feng, and Z. Shi, "SDSCNN: A Hybrid Model Integrating Static and Dynamic Spatial Correlation Neural Network for Traffic Prediction", *IEEE Access*, Vol. 10, pp. 121159-121172, 2022.

[18] C. Tang, J. Sun, Y. Sun, M. Peng, and N. Gan, "A General Traffic Flow Prediction Approach Based on Spatial-Temporal Graph Attention", *IEEE Access*, Vol. 8, pp. 153731-153741, 2020.

[19] S. Ke, Z. Pan, T. He, Y. Liang, J. Zhang, and Y. Zheng, "AutoSTG+: An automatic framework to discover the optimal network for spatio-temporal graph prediction", *Artificial Intelligence*, Vol. 318, p. 103899, 2023.

[20] X. Kong, W. Zhou, G. Shen, W. Zhang, N. Liu, and Y. Yang, "Dynamic graph convolutional recurrent imputation network for spatiotemporal

Received: September 1, 2023. Revised: October 7, 2023.

traffic missing data", *Knowledge-Based Systems*, Vol. 261, p. 110188, 2023.

[21] J. Qi, Z. Zhao, E. Tanin, T. Cui, N. Nassir, and M. Sarvi, "A Graph and Attentive Multi-Path Convolutional Network for Traffic Prediction", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 7, pp. 6548-6560, 2023.

[22] Z. Zhang, Y. Li, H. Song, and H. Dong, "Multiple dynamic graph-based traffic speed prediction method", *Neurocomputing*, Vol. 461, pp. 109-117, 2021.