Space-time Traffic Flow Prediction Scheme based on Graph Convolution Attention Nodes Influence

HUI ZHI+, YI-YUAN FENG AND MIAO-MIAO DUAN

Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University, Hefei 230601,China

School of Electronic and Information Engineering, Anhui University, Hefei 230601, China

E-mail: zhihui_0902@163.com;

Real-time and accurate Traffic Flow Prediction (TFP) can improve the efficiency of road traffic, and plays a crucial role in the field of intelligent transportation. Most of the existing TFP methods that consider the influence of non-adjacent nodes in space-time dimension rely on implicit training and learning from graph data to obtain the dependency relationships of non-adjacent nodes, lacking explicit definition of the physical propagation between non-adjacent nodes. Thus, insufficient training set data will directly affect the TFP accuracy, and the training and learning will lead to high computational complexity. In order to solve this problem, this paper proposes a space-time traffic flow prediction scheme based on graph convolution attention nodes influence (STFP-GCANI). The traffic network graph is defined according to the physical traffic road, and a graph attention node influence evaluation method (GANIEM) is designed to evaluate the influence value between different nodes (i.e., intersections) of traffic network graph without training and learning. The influence value between nodes is combined with the adjacency matrix, neighborhood matrix and corresponding free flow accessibility matrix to extract features of traffic flow graph. These features are input into a Graph Convolutional Attention Node Influence Long Short-Term Memory (GCANI-LSTM) network for space-time TFP. Simulation results show that, due to the explicitly modeling of mutual influence between non-adjacent nodes, the proposed STFP-GCANI can improve the accuracy of TFP and reduce training time.

Keywords: Space-time Traffic Flow Prediction, Deep Learning, Graph Attention Node Influence Evaluation Method

1. INTRODUCTION

In recent years, with the development of economy and the acceleration of the urbanization process, the number of motor vehicles in China has increased sharply, reaching 408 million [1]. Among them, there are 320 million cars, accounting for 78.4% of the total number of motor vehicles. With the significant increase in the number of motor vehicles, people are not only enjoying modern transportation convenience, but also suffering from traffic congestion.

Early traffic flow prediction (TFP) methods only focus on time series, resulting in lower prediction accuracy [2]. In order to solve this problem, some researchers have adopted deep learning methods to incorporate spatial dimension with time dimension, such as Convolutional Neural Network (CNN) [3-5], Graph Convolutional neural Network (GCN) [6-8] and attention mechanism [9-13]. However, these methods [3-13] only study the influence relationship between adjacent nodes in topology graphs, but ignore how congestion at a non-adjacent intersection affects the traffic flow, which may lead to the situation that the space-time TFP is limited to a local area, and failure to obtain the global traffic flow information, thus decreasing the prediction accuracy.

In order to solve this limitation, some studies have considered the influence of non-adjacent nodes in spatial-time dimension through attention mechanism [14-15] or implicit learning from graph data [16-18]. But these methods [14-18] rely on training and learning from graph data to obtain the dependency relationships of non-adjacent nodes, lacking a clear definition of the physical propagation paths of non-adjacent nodes. Therefore, insufficient training set data will directly affect the accuracy of predictions, and the training and learning will lead to high computational complexity. In addition, these methods do not incorporate easily obtainable physical information such as road length and free flow velocity as inductive biases into the model, resulting in poor generalization ability of the model.

Therefore, we consider explicitly modeling the mutual influence between non-adjacent nodes in space-time traffic flow prediction. The main contributions of this paper are as follows:

(1) A space-time traffic flow prediction method based on graph convolution attention nodes influence (STFP-GCANI) is proposed. In STFP-GCANI, the Graph Attention Node Influence Evaluation method

(GANIEM) is designed to evaluate the influence value of different nodes in topology graph. By combining these node influence values with neighborhood matrix and corresponding free flow accessibility matrix, the feature values of traffic flow are extracted. This explicit non-adjacent node impact assessment method does not rely on data training and learning, so STFP-GCANI can avoid the impact of insufficient training data on prediction accuracy and reduce computational complexity.

(2) In STFP-GCANI, a graph convolutional attention node influence long short-term memory (GCANI-LSTM) neural network is designed, and the extracted feature values are input into the GCANI-LSTM network for space-time traffic flow prediction. The road length and free flow velocity are also added into the model to improve the model's generalization ability.

This study aims to use deep learning to accurately predict the future traffic flow, which is of great significance for alleviating traffic congestion, improving road utilization efficiency, guiding urban traffic planning and optimizing public transport system.

2. RELATED WORK

With the increasing complexity of road networks in the process of urbanization, some early TFP researches that only consider the impact of time dimension, and do not consider the impact of spatial dimension and space-time combination dimension [2], will lead to low prediction accuracy. In order to solve this problem, some researchers adopt deep learning methods to incorporate spatial dimension, such as Convolutional Neural Network (CNN) [3-5], Graph Convolutional neural Network (GCN) [6-8] and attention mechanism [9-13].

Literature [3] uses CNN for space-time TFP, designs a space-time feature selection algorithm to screen the optimal data, and uses CNN learning to filter the data to extract features for prediction. During the research in [3], it is found that Recurrent Neural Network (RNN) variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), can be effectively used in space-time TFP due to their strong processing capacity for sequence data. They can effectively capture long-term dependencies in time series traffic data. In literature [4], based on CNN, the multi-gated convolutional neural network combined with CNNs and LSTM gets rid of the problem of low space-time correlation utilization of lower layers, and can simultaneously extract features from low to high levels and combine them with external factors and space-time dependence for space-time TFP.

GCN can obtain spatial dependency relationships by performing convolution operations on the basis of the constructed graph structure. Therefore, GCN can be used to construct the spatial TFP relationship between different nodes in the traffic road networks. Literature [6] proposes a dynamic space-time convolution layer based on superimposed graph structure. It can predict short-term space-time traffic flow by extracting features based on the relationship between time-dependent and space-time traffic flows and the dynamic changes of space-time graph structure, which is superior to CNN in accuracy. What's more, in order to capture the space-time dependence of traffic information simultaneously, a method of integrating RNN, LSTM, GRU and GCN models is proposed [6], which can make full use of the advantages of each model. Literature [7] considers the relationship between space and time in graph structures, adopts multi-stage dynamic and static fusion to enhance the extracted features, and uses diffusion graph convolution to build a model for timing and graph structure to better perform long and short-term TFP.

With the development of the attention mechanism in the direction of natural language processing, researches find that the attention mechanism makes the deep learning model more focused on important information, which can effectively improve the performance of model [9]. In literature [10], the attention mechanism is added in GCN, and an attention-based graph convolution network was designed to coordinate the spatial and temporal correlation. Combined with the coordinate method in the graph structure network, space-time TFP was carried out according to the influence degree of feature information in different dimensions.

Summarizing above researches on space-time TFP methods [3-13], we find that these methods only consider the relationship between the adjacent intersections in topological graphs and ignore the relationship between non-adjacent intersections in the TFP process, which may lead to the situation that the space-time TFP is limited to a local area, and failure to obtain the global traffic flow information, thus decreasing the prediction accuracy. Recent studies also have pointed out this limitation, for example, Yan et al. [19] explicitly state that "existing traffic flow prediction models struggle to explicitly model the space-time correlation between non-adjacent nodes." Shen et al. [20] also note in their survey that

“current graph neural network-based traffic prediction methods still face challenges in capturing long-range dependencies.” In real-world road networks, congestion at a non-adjacent bottleneck can propagate along traffic flows across multiple intersections and affect the traffic state of a target road segment within the free-flow travel time.

In order to solve this limitation, some studies have considered the influence of non-adjacent nodes through attention mechanism [14-15] or implicit learning from graph data [16-18]. For example, Graph Multi-Attention Network (GMAN) [14] adopts spatial and temporal attention mechanisms for non-adjacent nodes. Space-time Graph Attention Network (ST-GRAT) [15] introduces directional attention and a spatial sentinel. Diffusion Convolutional Recurrent Neural Network (DCRNN) [16] captures spatial dependencies via diffusion process. Augmented Multi-Component Recurrent Graph Convolutional Network (AMCGCN) [17] and Adaptive Graph Convolutional Recurrent Network (AGCRN) [18] implicitly learn similarity dependency relationships between different nodes through implicit learning of data in predefined and adaptive graphs.

Summarizing above literatures [14-18], we found that these studies of non-adjacent nodes have following issues. Firstly, these methods obtain the dependency relationships between non-adjacent nodes through attention mechanism or implicit learning from graph data. Although they can capture some spatial dependency relationships, their connection weights are completely learned from historical data, relying on the learning of the graph data. If there is a congestion pattern in the test set that has never been seen before during training, they may fail to correctly infer its impact. That is to say, these methods lack physical knowledge as support and lack explicit definition of the physical propagation path between non-adjacent nodes, insufficient training set data will directly affect the accuracy of prediction; Secondly, for long-distance (multi-hop) non-adjacent node pairs, their correlation in time series is often weak and sparse. These methods [14-18] require a large number of training samples to extract these weak relationships from noise, resulting in high computational complexity. Thirdly, in terms of physical prior information, these methods are purely data-driven and do not incorporate easily obtainable physical information such as road length and free flow velocity as inductive biases into the model, resulting in a decrease in the model's generalization ability when the data is sparse or the distribution is change.

To address these issues, we consider explicitly modeling the mutual influence between non-adjacent nodes and propose the space-time traffic flow prediction method based on graph convolution attention nodes influence (STFP-GCANI). In STFP-GCANI, the real urban traffic is mapped into the traffic network graph, and the neighborhood matrix and the corresponding free flow accessibility matrix are obtained from the traffic network graph. And then the GANIEM method is designed to evaluate the influence value of different nodes in the topology graph. These node influence values are combined with neighborhood matrix, and corresponding free flow accessibility matrix to perform flow graph convolution, thereby extracting the feature values of traffic flow. And then these extracted feature values are input into the GCANI-LSTM network for space-time TFP. The GANIEM is an explicit non-adjacent node impact assessment method that does not rely on data training and learning, but calculates by physical rules, so STFP-GCANI can avoid the impact of insufficient training data on prediction accuracy. Moreover, the explicit non-adjacent node impact assessment in GANIEM can significantly reduce computational complexity compared to implicit data training and learning of non-adjacent node impact in AMCGCN [17] and AGCRN [18], making it more suitable for traffic scenarios with a large number of nodes and complex road conditions. In addition, we incorporate physical prior information such as road length and free flow velocity into the GCANI-LSTM model, which can improve the model's generalization ability. Additionally, although AMCGCN [17] can model multi-level spatial dependencies, its “multi-level” emphasizes capturing different types of spatial relationships from multi type graph structures, while the “multi-level” in this paper focuses on the influence of nodes with different propagation hops on the physical road network, the two methods belong to different dimensions.

3. SYSTEMS MODEL

According to the roads and clients (sensors or other information collectors, i.e., the physical devices that sense traffic flow) in the real physical network, they are mapped to the traffic network graph to predict the space-time traffic flow based on the influence of attention nodes in the graph. The system model of the space-time TFP method based on graph convolution attention nodes influence (STFP-GCANI) is shown in Fig. 1.

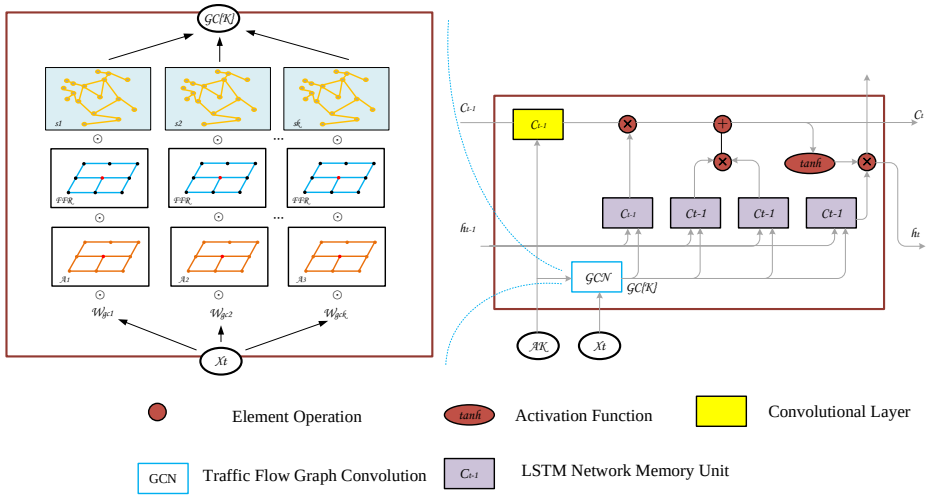


Fig. 1. System model of STFP-GCANI.

As show in Fig. 1, a traffic network graph based on traffic flow is constructed first, and the adjacency matrix and neighborhood matrix are obtained according to the traffic network graph, then the corresponding free flow accessibility matrix is set. Then, according to the influence between the graph nodes, graph attention node influence evaluation method (GANIEM) is designed to calculate the corresponding graph node influence evaluation value based on physical features. The features of traffic flow graph are obtained by traffic flow graph convolution based on the adjacency matrix, neighborhood matrix and corresponding free flow accessibility matrix. Finally, the features are input into the graph convolutional attention nodes influence long short-term memory (GCANI-LSTM) neural network to predict the space-time traffic flow.

3.1 Traffic Network Graph based on Traffic Flow

The traffic roads are bidirectional in reality, the impact of traffic congestion on traffic flow prediction is also bidirectional. Therefore, considering the factors of road bidirectional propagation [20], the traffic network graph is set as an undirected graph. The difference between the undirected graph of traffic network and molecular graph, social network graph is that: (1) the traffic state of each road in the traffic network is different in each time period; (2) Because the graph structure of the traffic network is fixed, there are very few isolated vertices or edges; (3) Each road in the traffic network has various physical characteristics, such as the number of lanes on different roads, the distance of roads and the speed limit of lanes. In addition, the traffic data of the road is collected by various clients (sensors or other information collectors). Most traffic data are measured based on location, but some traffic data may be measured based on the average state of the road and changes with time. Therefore, to ensure the stability of traffic flow graph structure, each node in the graph carries different traffic flow data, the clients (with sensors to sense the traffic flow) are taken as nodes, and the lines connecting different clients are taken as edges. Because the clients are usually set at intersections, thus the nodes are also means the intersections in this paper. The traffic network graph is represented as $G(V, E)$, where V represents the set of nodes $i \in V$, and E represents the set of edges, and $(i, j) \in E$ represents the edge between nodes (clients) i and j in the traffic network.

3.2 Adjacency Matrix and Neighborhood Matrix of Traffic Network Graph

In the traffic network graph G, there are N nodes, the connectivity is represented by the adjacency matrix $A \in R^{N*N}$. If nodes are connected between i and j, then $A_{i,j} = 1$; otherwise, nodes are connected without edges, then $A_{i,j} = 0$. Based on the adjacency matrix, it is known that the matrix of the number of adjacent edges of each node is $D \in R^{N*N}$, where $D_{i,i} = \sum_j A_{i,j}$. D is a diagonal matrix, all elements except diagonal are 0. Based on the adjacency matrix, the edge function from node i to node j is $d(i, j)$, and each node has adjacent nodes. Therefore, the *k-hop neighborhood set* of the node i is defined as $\{d(i, j) \le k \mid i, j \in V\}$. Due to the fact that the collected traffic flow data is temporal, the current traffic flow situation will affect the prediction of future traffic flow, and it is believed that each road in the traffic network is self-influencing. Therefore, the one hop neighborhood of a node in the traffic network graph

includes the node itself and a neighborhood matrix, which is

$$\tilde{A} = A + I \tag{1}$$

where I is identity matrix. According to the one hop neighborhood set, we can obtain the *k-hop neighborhood* set of the node *i*, which is $(A+I)^k$. The *k-hop* neighborhood relationship of node *i* in the traffic network graph is only used to describe the existence of surrounding *k-hop* neighborhood nodes. As the elements of $(A+I)^k$ are inevitably greater than 1, we limit the element values of $(A+I)^k$ to the (0,1) interval, and the *k-hop neighborhood* nodes are not weighted by the number of hops. Therefore, define a *k-hop neighborhood matrix* $\tilde{A}^k$ as follows,

$$\tilde{A}_{i,j}^k = \min\left((A+I)_{i,j}^k, 1\right) \tag{2}$$

where $\min(\cdot)$ represents the minimum value that ensures the description of *k-hop* neighborhood relationships. $\tilde{A} = A^1 = A$, as show in figure 1, the center node is represented by red dot, the *k-hop* neighborhood nodes are represented by orange dots.

3.3 Free Flow Accessibility Matrix

In the traffic network graph, roads are mapped to edges, intersections are mapped to nodes, there are different lengths of edges. The distance matrix is defined as $D \in R^{N*N}$, where elements $D_{i,j}$ represent the distance from node *i* to node *j*, i.e., the length of this road. When considering the physical characteristics of the traffic network, there are mainly two types of impacts between different roads: (1) The deceleration, acceleration, and road congestion during driving towards another road; (2) The behavior choices of drivers and the different characteristics of vehicles during driving towards another road. Therefore, when considering the interaction between non-adjacent nodes, intermediate nodes cannot be ignored, and the accessibility between these nodes should be considered. The traffic flow between *k-hop* adjacent nodes should follow existing theories [21], and the free flow accessibility analysis is as follows:

$$FFR_{i,j} = \begin{cases} 1, & S_{i,j}'^{FF}\beta\Delta t - D_{i,j} \geq 0 \\ 0, & S_{i,j}'^{FF}\beta\Delta t - D_{i,j} < 0 \end{cases} \tag{3}$$

where $FFR \in R^{N*N}$ is the free flow accessibility analysis function, $S_{i,j}'^{FF}$ represents the average speed (i.e., free flow speed) from node *i* to node *j in the* traffic network graph under clear weather or non-congested roads. Δt indicates the time interval. β is the time step size, it means the time intervals considered over a distance based on free flow speed, and determines the impact on free flow accessibility. If it is possible to drive from node *i* to node *j* at free flow speed $S_{i,j}'^{FF}$ within time $\beta\Delta t$, thus $FFR_{i,j} = 1$, otherwise $FFR_{i,j} = 0$. Therefore, $FFR_{i,j}$ can be used to measure whether a vehicle in the traffic network graph can move from node *i* to node *j* at free flow speed within a specific time. In traffic network graph, assume that any two nodes are accessible. The blue line in Figure 1 represents the free flow accessibility analysis FFR.

3.4 Node Influence Evaluation Method

There are large number of nodes in the traffic network graph, not every node has a significant impact on the accuracy of TFP. Due to the differences of intersections, such as different number of roads merging into main road, different location of intersections, and different distances between intersections, the impact of source_intersection to each intersection on TFP varies. Therefore, a graph attention node influence evaluation method (GANIEM) is proposed to evaluate the influence value of source intersection (node) on each intersection's TFP prediction accuracy.

The core idea of GANIEM is to answer the question "When a traffic jam occurs at an intersection, which other intersections will be affected and to what extent?". Traditional graph convolution only considers directly connected intersections (1-hop neighbors), but in reality, the impact of congestion will propagate along the road to further intersections (2-hop, 3-hop...). However, not all of these intersections will be equally affected. If the distance between two intersections is far or the speed limit on the road is low, congestion is difficult to pass through in a short period of time. GANIEM quantifies this impact through two steps. The first step is to identify all intersections within the k-hop range around the source intersection based on the road network topology. The second step is to determine whether congestion can truly propagate from the source intersection to the target intersection within a given time window when taking into account the road length and free flow velocity. Only intersections that meet both the

conditions of "topologically within k hops" and "reachable within free flow time" will be assigned influence values. The influence value is calculated according to expression (5) and does not rely on training data, thus providing reliable physical prior information for GCANI-LSTM model.

Firstly, the adjacency matrix A can be used to calculate the number of connections between each node and other nodes within k hops [22]. $L^k = \Sigma_{i=1}^{k} A^i$ with element l_{ij}^1 represents the number of hops from node i to node j. The larger the l_{ij}^1, the better the connectivity. That is, if the intersection mapped to the current graph node experiences congestion, it has great impact on other intersections. Finally, based on connectivity and adjacency matrix, the influence of node on the prediction accuracy is evaluated by the calculating based on one hop neighborhood matrix. The expressions are as follows

$$\tilde{n} = \frac{\exp(W^n)}{\sum_{i=1}^{N} \exp(W^i)} \tag{4}$$

$$s = \sigma\left(r\hat{A}\tilde{n}\right) \tag{5}$$

where the one hop neighborhood matrix is $\hat{A} = D^{-1/2}\tilde{A}D^{-1/2}$, $\tilde{A} = A + I$, σ is the activation function, the influence evaluation ratio is $r = L_i^1 / \Sigma_{i=1}^{N} L_i^1 \in (0,1]$, W is the weight matrix, s represents the node influence evaluation value, and $\tilde{n}$ represents the adding softmax activation function to the attention mechanism based on the weight W. Mapping the node influence evaluation value output to the (0,1) interval, and then doing weight calculation. Computing based on expression (5), the goal of automatically learning the node influence value parameters can be achieved.

From expressions (4) and (5), we can see that GANIEM first evaluates node connectivity using L^k, then computes the influence value via $s = \sigma(r \cdot \tilde{A})$, and finally normalizes the attention weights using softmax. The influence value of each graph node obtained by above computation is shown in Fig. 2.

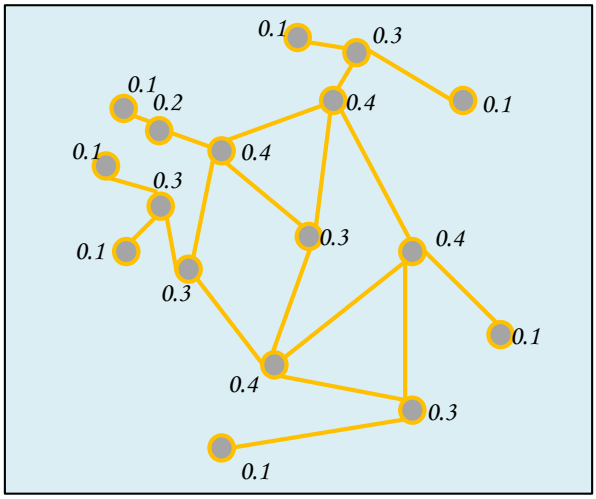


Fig. 2. Influence value of each graph node using GANIEM.

3.5 Traffic Flow Prediction Problem

Usually, traffic flow prediction mainly relies on collecting traffic status data (such as traffic flow, traffic speed, and vehicle travel time) during a certain time period to predict future traffic status. In this section, real-world traffic is mapped to a traffic network graph, which consists of N nodes. Edges are mapped by roads between nodes. During a certain period of time, the clients (sensors) at the locations of these nodes collect traffic flow data, represented by $x_t \in R^N$.

The purpose of solving the problem of traffic flow prediction is to learn the $F(\cdot)$ function of the time interval T of historical signals in the traffic network graph, that is, the $X_T = [x_1, \cdots, x_t, \cdots, x_T]$ is traffic flow status information obtained during one or more time steps. In this paper, we mainly focus on the predicting of future x_{T+1} traffic status information under historical space-time traffic information, x_{T+1} traffic flow prediction is

$$F\left(\left[x_1, \cdots, x_t, \cdots, x_T\right]; G\left(V, E, \tilde{A}^k, FFR, s\right)\right) = x_{T+1} \tag{6}$$

Meanwhile, we also investigate the influence of weights and weight parameters between nodes in the

functions $F(\cdot)$ in TFP.

4. SPACE-TIME TRAFFIC FLOW PREDICTION METHOD BASED ON GRAPH CONVOLUTION ATTENTION NODES INFLUENCE (STFP-GCANI)

4.1 Traffic Flow Graph Convolution

In previous studies [23-25], graph convolutional network architecture was designed based on adjacency matrix. Typically, the regions in the input matrix that affect the convolutional results were called receptive fields, while the purpose of convolutional networks was to extract local features from the input matrix structure of the convolutional layer. Therefore, the core idea of graph convolutional layer is to extract local features from the traffic flow data that input into the traffic network structure. Therefore, the graph W convolution operation of a one hop neighborhood is defined as feature extraction by multiplying the input data x_t with the one hop neighborhood matrix $\tilde{A}$ and the trainable weight matrix W, i.e., $\tilde{A}x_tW$. The scope of the graph convolution operation of current one hop is limited by the adjacent one hop nodes. The existing researches that consider the influence of non-adjacent nodes all use training and learning from graph data to obtain the dependency relationships of non-adjacent nodes. There is no free reachability analysis mapped to edges, and the high-order neighborhood analysis of nodes. Therefore, in order to obtain explicitly modeling of non-adjacent node influence and consider the issue of edge attributes, a *k-hop neighborhood matrix* $\tilde{A}^k$ is used instead of a one hop neighborhood matrix $\tilde{A}$ to define the *k-hop* influence evaluation and traffic flow graph convolution features. The expressions are as follows.

$$s^k = \sigma\left(r\hat{A}^k W\right) \tag{7}$$

$$GC_t^k = \left(W_{gc_k} \odot \tilde{A}^k \odot FFR \odot s^k\right) x_t \tag{8}$$

where $\odot$ is the multiplication operator of matrix elements. From one hop influence value $s = \sigma\left(r\hat{A}\tilde{n}\right)$, we can obtain k hop influence value s^k. $W_{gc_k} \in R^{N*N}$ is the trainable weight matrix of the k hop traffic network graph convolution, $x_t \in R^N$ is the input vector of the traffic state (traffic flow) of the traffic network graph, GC_t^k is the graph convolution feature value that extracted from the k hop nodes of the traffic network graph at time *t*. Due to the fact that both free flow accessibility matrix FFR and k hop neighborhood matrix $\tilde{A}^k$ contain only sparse elements with 0 and 1, the GC_t^k is also a sparse matrix. At the same time, the trainable weight matrix W_{gc_k} and graph node influence evaluation value s^k can measure the interaction between graph nodes, improving the prediction accuracy of TFP graph convolutional model. k is a positive integer, the larger k, the larger convolution range of the traffic network graph, the more features are extracted based on neighboring nodes. For example, for a specific graph, as k is gradually increases, $\tilde{A}^k \odot FFR$ will converge to FFR, i.e., when $k = k_{max}$, $\tilde{A}^{k_{max}} \odot FFR = FFR$. Therefore, in order to solve the TFP problem, k cannot be infinitely large, that is, k is not necessarily set as k_{max}. At the same time, it is necessary to consider the balance between rich features and prediction accuracy.

The number of hops in the neighborhood x_t is different, thus the feature extracted are also different. The maximum number of hops in traffic network graph convolution is set to $k \le k_{max}$, and the traffic network graph convolution feature corresponding to the input traffic flow data is GC_t^k. In order to increase the feature space, the convolutional feature set of different hop numbers in the traffic network graph is formed into a vector, which is

$$GC_t^{\{K\}} = \left[GC_t^1, GC_t^2, \cdots, GC_t^k\right] \tag{9}$$

where $GC_t^{\{K\}} \in R^{N*N}$ contains convolutional features of all jumps from 1 to k, x_t is obtained from the input data through traffic network graph convolution, $GC_t^{\{K\}}$ will be used as input to the LSTM neural network for prediction.

4.2 Graph convolutional attention node influence long short-term memory (GCANI-LSTM) neural network

In this section, a graph convolutional attention node influence long short-term memory (GCANI-LSTM) neural network is designed to learn dynamic temporal dependencies and complex spatial dependencies in traffic flow data. In this model, the input is feature vectors $GC_t^{\{K\}}$, but the hidden state and gate structure in LSTM remain unchanged. The expressions for input door, forget door and output door and input unit states in time step $t = \varpi\beta$ (ϖ is a multiple) are as follows respectively.

$$i_t = \sigma_g\left(W_i \cdot GC_t^{\{K\}} + U_i \cdot h_{t-1} + b_i\right) \tag{10}$$

$$f_t = \sigma_g\left(W_f \cdot GC_t^{\{K\}} + U_f \cdot h_{t-1} + b_f\right) \tag{11}$$

$$o_t = \sigma_g\left(W_o \cdot GC_t^{\{K\}} + U_o \cdot h_{t-1} + b_o\right) \tag{12}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot GC_t^{\{K\}} + U_C \cdot h_{t-1} + b_C\right) \tag{13}$$

where $\cdot$ is the matrix element multiplication operator, W_i、W_f、W_o and $W_C \in R^{N*N}$ are weight matrices, U_i、U_f、U_o and $U_C \in R^{N*N}$ are hidden state weight matrices, σ_g is activation function, $\tanh$ is hyperbolic tangent function, b_i、b_f、b_o and $b_C \in R^N$ are bias vectors, $GC_t^{\{K\}}$ are mapped into input unit state and three gates.

Due to the influence of adjacent and non-adjacent nodes in the traffic network, the current state of nodes in GCANI-LSTM is also affected by the previous LSTM network state of adjacent and non-adjacent nodes. Therefore, adding an attention mechanism and a unit state gate to LSTM can make GCANI-LSTM focus on the previous LSTM network state of adjacent and non-adjacent graph nodes better.

$$\alpha_t = \frac{\exp(C^n)}{\sum_{i=1}^{N}\exp(C^i)} \tag{14}$$

$$C_{t-1}^* = W_N \odot \left(\tilde{A}^k \odot FFR \odot s^k\right) \cdot \alpha_t C_{t-1} \tag{15}$$

where α_t indicates that the attention mechanism adds a softmax activation function, which maps the network state output to the interval (0,1) to calculate the weight. During the training process of GCANI-LSTM network, the goal of automatically learning weight parameters is achieved. W_N is the weight matrix of adjacent graph nodes, $\tilde{A}^k \odot FFR \odot s^k$ is used to constrain W_N and reflect the structure of the traffic network graph. When passing through this unit state gate, the influence of the current and previous unit states of adjacent units on subsequent time steps should be considered to obtain the final and hidden states. Thus,

$$C_t = f_t \odot C_{t-1}^* + i_t \odot \tilde{C}_t \tag{16}$$

$$h_t = o_t \odot \tanh(C_t) \tag{17}$$

After the final time step t, the hidden state h_t obtained through GCANI-LSTM is the predicted value $\hat{y}_t = h_t$, and the label of input data x_t is represented by y_t, so for the traffic flow prediction of the next time step $t+1$, the input is label y_t of time step t, that is, $x_{t+1} = y_t$. The loss function in the GCANI-LSTM process is

$$Loss = L\left(y_t, \hat{y}_t\right) = L\left(x_{t+1}, h_t\right) \tag{18}$$

where $L(\cdot)$ is the loss function between the predicted value of GCANI-LSTM and the true value, and the error value is generally calculated using MSE.

4.3 Algorithm of STFP-GCANI

In order to better describe the proposed space-time traffic flow prediction scheme based on graph convolution attention nodes influence (STFP-GCANI), the algorithm of STFP-GCANI is given in algorithm 1. Using the collected traffic flow state data and the related graph matrix, the feature values

are obtained as the input of GCANI-LSTM, and the final forecast output h_t is obtained after a certain time step. In this paper, the process of small-batch gradient descent and the parameter update process of backpropagation are simplified in the pseudo-code.

Algorithm 1 STFP-GCANI

Input: Traffic flow data $X_T = [x_1, \cdots, x_t, \cdots, x_T]$, Neighborhood matrix $\tilde{A}^1, \cdots, \tilde{A}^k$, Free flow accessibility matrix FFR, Node influence evaluation value s^k according to GANIEM

Output: Hidden Status h_t, that is, the predict output value $\hat{y}_t$

Parameter: weight W_{gc_k}、U_S、W_S、W_N and bias b_S

Initialize: $h_0 = 0 \in R^N$, $C_0 = 0 \in R^N$

GCN:

for t=1 to T **do**

for k=1 to K **do**

$GC_t^k \leftarrow \left(W_{gc_k} \odot \tilde{A}^k \odot FFR \odot s^k\right) x_t$

$GC_t^{\{K\}} \leftarrow \left[GC_t^1, GC_t^2, \cdots, GC_t^k\right]$

end

GCANI-LSTM:

C_t、h_t =GCANI-LSTM $(x_t, GC_t^{\{K\}}, C_{t-1}, h_{t-1})$

end

5. SIMULATION EXPERIMENT AND ANALYSIS

5.1 Experimental Data Sets and Environmental Parameters

In this section, based on the data sets of highway PeMS traffic data provided by Caltrans, two open data sets, PeMSD4 and PeMSD8, were selected for traffic flow prediction to evaluate the performance of STFP-GCANI scheme in this chapter. These two data sets are from the California's San Francisco Bay Area and SAN Bernardino County. These data sets record traffic flow, average occupancy, and average speed in the fixed area. The data set details are shown in table 1.

The PeMS system collects data every 30 seconds, that's to say, PeMS uploads real-time data every 30 seconds, and due to the aggregation every 5 minutes, there are 12 recorded data every hour. The time steps are 16992 and 17856 respectively.

There are two types of missing values in the raw data, which are consecutive missing segments caused by temporary sensor failures, and random single-point missing values. The overall missing rates of PeMSD4 and PeMSD8 are approximately 3.2% and 4.5% respectively. We use linear interpolation to fill the missing values, for a consecutive missing segment of length L, the two nearest valid observations (v_{start} and v_{end}) that before and after the segment are used to perform linear interpolation in chronological order

$$v_t = v_{start} + (v_{end} - v_{start}) \times \frac{t - t_{start}}{t_{end} - t_{start}}, t = t_{start} + 1, \ldots, t_{end} - 1 \quad (19)$$

For missing values at the beginning or end of the data sequence, forward or backward filling is applied using the nearest valid observation.

To eliminate scale differences among different sensors, we apply min-max normalization to the traffic flow series of each sensor separately,

$$\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (20)$$

where $x_{\min}$ and $x_{\max}$ are the minimum and maximum flow values in the training set of the sensor,

respectively. The test set is obtained by normalizing the parameters of the training set to ensure data consistency. After the predicted outputs are transformed to the original flow values through inverse transformation, evaluation metrics such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are calculated.

Table 1. Experimental Data Sets

Data Set	Time Horizon	Number of sensors	Number of edges	Time steps
PeMSD4	2018/1/1-2/28	307	340	16992
PeMSD8	2016/7/1-8/31	170	295	17856

In simulations, the STFP-GCANI model is build based on PyTorch deep learning framework with model construction and training environment setting, and the code is written in PyCharm. The graph convolution hop number is set as $K = 3$ to evaluate and compare experimental models, and the hidden state dimension is the number of nodes in the traffic network graph. Adam is selected as the optimizer to solve the problem of gradient explosion and disappearance, and the initial learning rate is 10^{-4} and the batch size is 16 to train the GCANI-LSTM model. Three indexes were used to evaluate the proposed model in space-time TFP, which are MAE, MAPE and RMSE, the calculation methods are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_t - \hat{y}_t\right| \tag{19}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| * 100\% \tag{20}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_t - \hat{y}_t\right)^2} \tag{21}$$

5.2 Analysis of Experimental Results

We compare the simulation results of STFP-GCANI with that of following TFP methods, comparison results are shown in Table 2, and the explanations of comparison methods are as follows.

(1) Auto-regressive integrated moving average (ARIMA): ARIMA is one of the most classical methods for predicting time series of traffic flow.

(2) Support vector regression (SVR): Support vector machine uses the linear relationship between traffic flow input and output to train the model.

(3) Convolutional long short term memory neural network (Conv-LSTM): Using convolution operation to replace the connected part of the LSTM, the output depends not only on the input but also on the partial past state, which can predict the traffic flow more accurately.

(4) Local spectral graph convolution long short term memory neural network (LSGC-LSTM): This model does not need feature decomposition, only focuses on parameter k, and has good spatial localization to extract weighted features for TFP.

(5) DCRNN (Diffusion Convolutional Recurrent Neural Network): This model captures spatial dependencies via diffusion process.

(6) AGCRN (Adaptive Graph Convolutional Recurrent Network): This model learns node-specific patterns and adaptive graphs.

(7) GMAN (Graph Multi-Attention Network): This model uses spatial and temporal attention mechanisms.

(8) ST-GRAT (Spatio-Temporal Graph Attention Network): This model introduces directional attention and a spatial sentinel.

(9) AMCGCN (Augmented Multi-Component recurrent Graph Convolutional Network): This model fuses multiple predefined graphs and adaptive graph for spatial dependency modeling.

(10) Space-time traffic flow prediction scheme based on graph convolution attention nodes influence (STFP-GCANI): The traffic graph convolution operator is used to extract features, and the attention node influence evaluation method is used to pay attention to the complex temporal and spatial dynamic relationship in graph convolution LSTM network.

Table 2. Performance comparison of different TFP methods.

Model	PeMSD4			PeMSD8		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
ARIMA	28.24	57.23%	40.35	22.24	40.43%	31.49
SVR	29.45	58.34%	44.85	23.43	42.23%	33.91
Conv-LSTM	23.08	44.76%	34.55	18.91	35.57%	27.92
LSGC-LSTM	22.52	50.87%	33.41	18.99	45.56%	26.95
DCRNN	20.62	40.15%	29.53	16.38	30.12%	24.81
AGCRN	20.11	39.28%	28.97	15.94	29.37%	24.35
GMAN	20.03	39.01%	28.85	15.86	29.14%	24.22
ST-GRAT	19.87	38.83%	28.64	15.72	28.91%	24.05
AMCGCN	19.72	38.71%	28.49	15.61	28.76%	23.93
STFP-GCANI	19.55	38.62%	28.22	15.47	28.45%	23.75

As shown in Table 2, different TFP methods are compared based on two data sets, PeMSD4 and PeMSD8, it can be seen that the proposed STFP-GCANI is superior to all other methods. Traditional methods (ARIMA, SVR) are linear and cannot capture complex space-time dependencies, resulting in the highest prediction errors. Conv-LSTM uses convolutional operations to replace fully connected structures in LSTM, which helps extract local temporal features, but it lacks explicit spatial graph modeling. LSGC-LSTM applies a polynomial-filtered local spectral graph convolution, which focuses on a fixed number of hops (parameter k) but ignores other important network characteristics, leading to suboptimal accuracy. Recent advanced space-time models (DCRNN, AGCRN, GMAN, ST-GRAT, AMCGCN) capture spatial dependency relationships through implicit learning from graph data, without explicit definition of the physical propagation between non-adjacent nodes. In contrast, the proposed STFP-GCANI explicitly models the influence of non-adjacent nodes by k-hop neighborhood matrix, free flow accessibility matrix and GANIEM method, without the need for training and learning, so it can reduce the impact of insufficient training data on prediction accuracy.

Fig. 3 shows the effect of graph convolution hop number on loss function in STFP-GCANI. When the hop number k increases, the error of PeMSD4 and PeMSD8 data sets are decrease gradually and eventually flatten out. Since the PeMSD8 data set is not corresponding a gentle road, it contains various road types and complex physical characteristics, and the data speed in PeMSD8 is assumed to be an integer. Therefore, the error of PeMSD8 data set is smaller than that of PeMSD4 data. It is easy to see from the figure 3 that, when $k>2$, the prediction error (loss function) is almost stays at the same level. Therefore, in table 2, we select $k=3$ in STFP-GCANI to compare STFP-GCANI with other TFP methods.

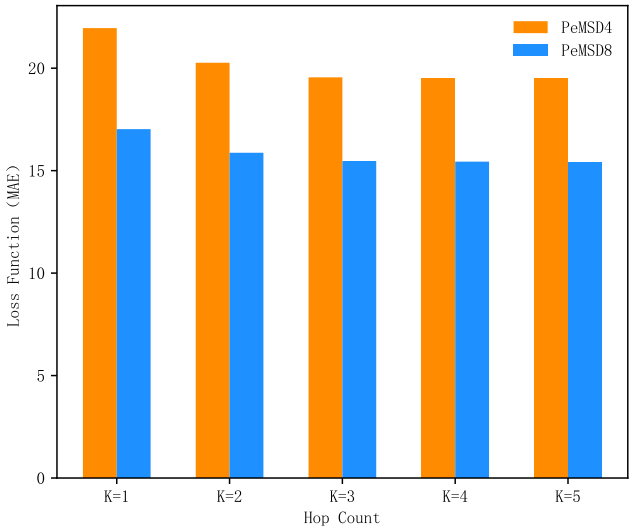


Fig. 3. Effect of graph convolution hop number on loss function in STFP-GCANI

The effect of graph attention node influence evaluation method (GANIEM) on the prediction performance is tested by experiments. In Fig.4, Fig.5 and Table 3, node influence value was evaluated by GANIEM, clustering method, random method and without GANIEM respectively. The clustering method selects some nodes (50% of the graph nodes) according to the aggregation relationship between nodes, groups these nodes into high-degree and low-degree clusters based on node degree, and assign fixed influence values of 0.8 and 0.2, respectively. The random method replaces s^k with uniformly random values, and ensure the fairness of each graph node's influence value within the mean value range ± 0.1. The without GANIEM replaces s^k with an identity matrix, which means that the influence between nodes is not considered. In order to facilitate the test and ensure the fairness of prediction to the maximum extent possible, a part of the graph nodes of the traffic network in Fig. 4 are used, and the calculated influence values of these nodes are compared among GANIEM, clustering method and random method. It can be seen from Fig. 4 that, the high evaluation values of nodes in the clustering method coincide with the high evaluation values of nodes in GANIEM to a certain extent. However, GANIEM takes into account the influence of connectivity between nodes. Therefore, in Fig. 5, the loss function of GANIEM is lower than that of other two methods, and in Table 3, the prediction error of GANIEM is smaller than that of other methods. This indicates that the explicit modeling node influence in GANIEM can improve TFP prediction accuracy. In addition, it can be seen from Table 3 that the improvement on PeMSD8 is greater, indicating that GANIEM plays a more significant role in networks with more complex road topologies.

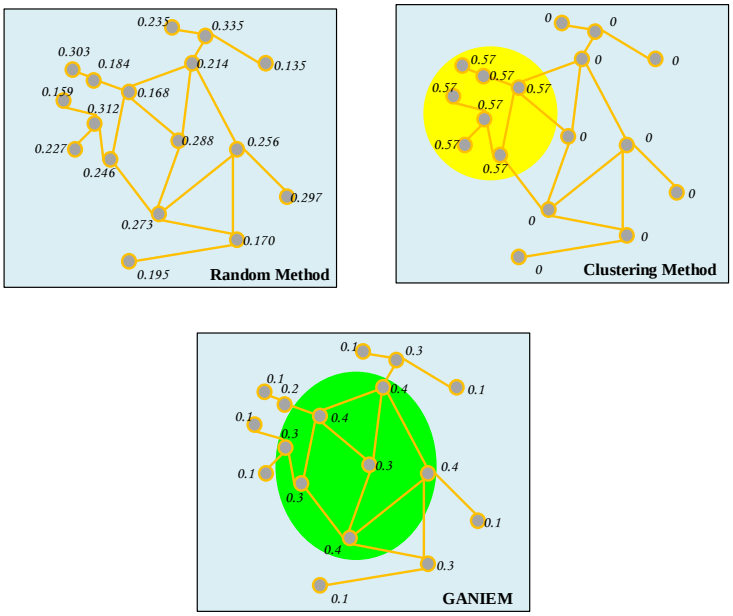


Fig. 4. Comparison of node influence evaluation values for GANIEM, clustering and random method

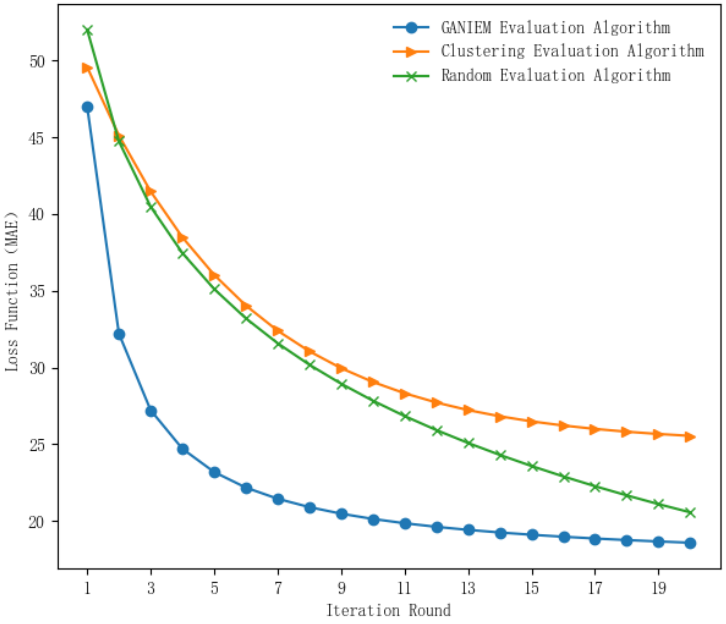


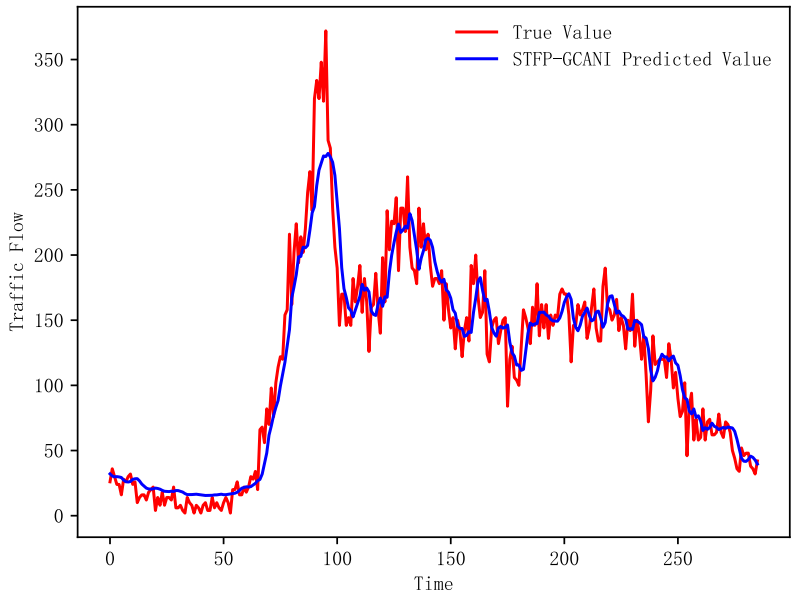
Fig. 5. Comparison of loss function for GANIEM, clustering and random method

Table 3. Experimental results of GANIEM on the PeMSD4 and PeMSD8 datasets

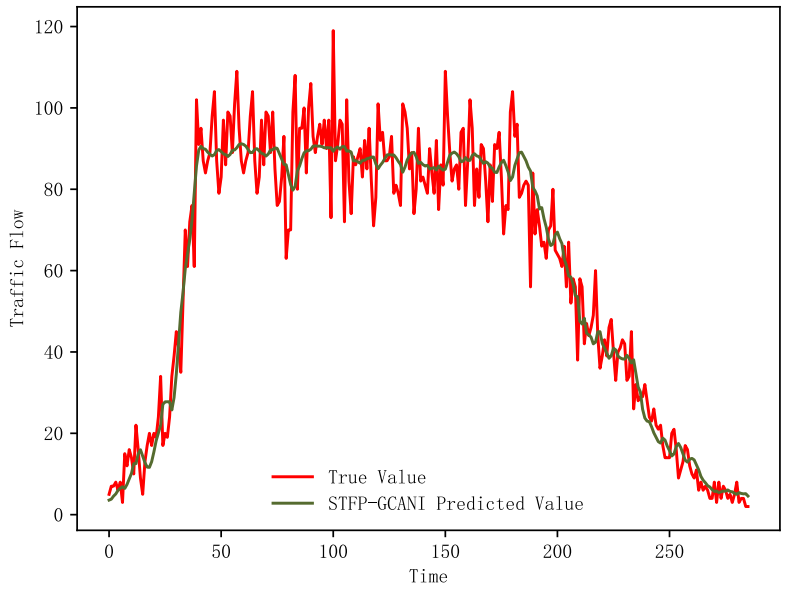
Different node influence evaluation methods	PeMSD4 MAE	PeMSD8 MAE
GANIEM (i.e., STFP-GCANI)	19.55	15.47
without GANIEM	20.83	16.97
Random influence	20.41	16.45
Clustering-based influence	20.12	16.21

Figs. 6 (a) and (b) show the comparison of STFP-GCANI prediction value and real value in two data sets. A location is selected from PeMSD4 and PeMSD8 data sets respectively and STFP-GCANI is used to predict the traffic flow. As shown in Figs. 6 (a) and (b), although the physical structure of the two traffic network graphs is different, the prediction value is very close to the real value during peak and

off-peak hours, so the prediction effect of STFP-GCANI is good.



（a） Comparison of prediction value and real value in PeMSD4 data set



（b）Comparison of prediction value and real value in PeMSD8 data set

Fig. 6. Comparison of STFP-GCANI prediction value and real value in two data sets

To further validate the performance improvement of our method, we conduct a statistical significance testing (Paired t-test) to confirm that the differences between our method and the compared methods are statistically robust and not due to random variation. The paired t-test (one-sided, α=0.05) was performed on the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) at each time step in the test set. The results are shown in Table 4, each p-value means the differences between the compared method and STFP-GCANI, we can see that all the p-values are below 0.05 on PeMSD4 and PeMSD8 datasets, this indicates that the improvements of the proposed method in terms of MAE, MAPE, and RMSE are statistically robust and not caused by

random fluctuations.

Table 4. Paired t-test results of different TFP methods

Dataset	TFP methods	p-value (MAE)	p-value (MAPE)	p-value (RMSE)
PeMSD4	DCRNN	0.004	0.006	0.003
PeMSD4	AGCRN	0.021	0.025	0.018
PeMSD4	GMAN	0.012	0.015	0.010
PeMSD4	ST-GRAT	0.008	0.011	0.007
PeMSD4	AMCGCN	0.019	0.022	0.016
PeMSD8	DCRNN	0.001	0.002	0.001
PeMSD8	AGCRN	0.023	0.027	0.019
PeMSD8	GMAN	0.010	0.013	0.009
PeMSD8	ST-GRAT	0.006	0.007	0.005
PeMSD8	AMCGCN	0.015	0.018	0.013

To demonstrate that the proposed node influence evaluation method GANIEM can reduce computational complexity, we compare their training time (seconds per epoch) of different TFP model on the PeMSD4 and PeMSD8 datasets under the same hardware configuration (NVIDIA Tesla V100 GPU, batch size=16). The results are shown in Table 5.

Table 5. Training time comparison of different TFP models (seconds per epoch)

Different TFP Models	PeMSD4(s/epoch)	PeMSD8(s/epoch)
DCRNN	39.7	44.6
AGCRN	34.6	38.9
GMAN	59.7	67.1
ST-GRAT	64.0	71.9
AMCGCN	74.5	83.8
STFP-GCANI	25.8	29.0

From Table 5, it can be seen that the training time of our proposed STFP-GCANI is significantly lower than all other models on both PeMSD4 and PeMSD8 datasets. On PeMSD4, STFP-GCANI requires only 25.83 seconds per epoch, which is about 35% lower than DCRNN, 25% lower than AGCRN, and 57%, 60%, and 65% lower than GMAN, ST-GRAT, and AMCGCN, respectively. On PeMSD8, the training time of our model is also much lower than that of other models. This efficiency advantage is mainly because that the k-hop adjacency matrix, free-flow accessibility matrix and node impact value in GANIEM are both pre-computed, which do not require training and learning for spatial relationships, resulting in low computational complexity. In contrast, all other models obtain the dependency relationships of non-adjacent nodes based on training and learning from graph data, resulting in high computational complexity. For example, DCRNN and AGCRN require iterative diffusion processes or adaptive graph training, GMAN and ST-GRAT use full-pair attention training with $O(N^2)$ complexity, and AMCGCN needs training from multiple predefined graphs and an adaptive graph.

6. SUMMARY

This paper presents a space-time traffic flow prediction scheme based on graph convolution attention nodes influence (STFP-GCANI). Firstly, a graph structure is used to represent the traffic network, and the graph attention node influence evaluation method (GANIEM) is used to evaluate the influence value of graph node. Then, the local features are extracted by combining the free flow accessibility matrix and adjacent matrix, and the temporal and spatial features are captured by the flow graph convolution operation of adjacency matrix, neighborhood matrix and corresponding free flow accessibility matrix, influence value between nodes. Finally, the graph convolutional attention node influence long short-term memory (GCANI-LSTM) neural network is designed to predict the traffic flow in the traffic network graph. The experimental results of two data sets show that the proposed STFP-GCANI method can improve the accuracy of TFP and reduce computational complexity, due to its explicitly modeling of

mutual influence between non-adjacent nodes. Due to space and time constraints, this paper did not provide a complete theoretical analysis of computational complexity and scalability, which will be done in our future work.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (62001001), Key Research and Development Plan in High tech Field of Anhui Province (202304a05020011), Graduate Education Quality Engineering Project of Anhui Province (2023qygzz005), Education Quality Engineering Project of Anhui University(2024xjzlgc447).

References

1. Yang Huihui. “Research on short-term traffic flow prediction method of urban road,” *Henan Science and Technology*, 2018, pp. 120-121.
2. Li F., Feng J., Yan H., et al., “Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, Vol. 17, 2023, pp. 1-21.
3. W. Zhang, Y. Yu, Y. Qi, et al., “Short-term Traffic Flow Prediction based on Spatio-Temporal Analysis and CNN Deep Learning,” *Transport metrica A-Transport Science*, Vol. 15, 2019, pp. 1688-1711.
4. C. Chen, K. Li, S. G. Teo, et al., “Citywide Traffic Flow Prediction Based on Multiple Gated Spatio-temporal Convolutional Neural Networks,” *ACM Transactions on Knowledge Discovery Data*, Vol. 14, 2020, pp. 42.
5. Y. Cheng, W. Xing, W. Pedrycz, et al., “NFIG-X: Nonlinear Fuzzy Information Granule Series for Long-Term Traffic Flow Time-Series Forecasting,” *IEEE Transactions on Fuzzy Systems,* Vol. 31, 2023, pp. 3582-3597.
6. K. Chen, F. Chen, B. S. Lai, et al., “Dynamic Spatio-Temporal Graph-Based CNNs for Traffic Flow Prediction,” *IEEE Access*, Vol. 8, 2020, pp. 185136-185145.
7. Z. Cui, J. Zhang, G. Noh, et al., “MFDGCN: Multi-stage Spatio-temporal Fusion Diffusion Graph Convolutional Network for Traffic Prediction,” *Applied Sciences-Basel*, Vol. 12, 2022, pp. 2688.
8. F. Rempe, P. Franeck, K. Bogenberger, “On the Estimation of Traffic Speeds with Deep Convolutional Neural Networks Given Probe Data,” *Transportation Research Part C-Emerging Technologies*, Vol. 134, 2022.
9. G. Brauwers, F. Frasincar, “A General Survey on Attention Mechanisms in Deep Learning,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, 2023, pp. 3279-3298.
10. M. Li, M. Li, B. Liu, et al., “Spatio-Temporal Traffic Flow Prediction Based on Coordinated Attention,” *Sustainability*, Vol. 14, 2022, pp. 7394.
11. L. Zhao, Y. Song, C. Zhang, et al., “T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, 2020, pp. 3848 – 3858.
12. M. Méndez, M. G. Merayo, M. Nunez, “Long-Term Traffic Flow Forecasting Using a Hybrid CNN-BiLSTM Model,” *Engineering Applications of Artificial Intelligence*, Vol. 121, 2023.
13. Q. Zhang, M. Tan, C. Li, et al., “Spatio-Temporal Residual Graph Convolutional Network for Short-Term Traffic Flow Prediction,” *IEEE Access*, Vol. 11, 2023, pp. 84187-84199.
14. C. Zheng, X. Fan, C. Wang, and J. Qi, “GMAN: A graph multi-attention network for traffic prediction,” *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34, 2020, pp.

1234-1241.
15. C. Park, C. Lee, H. Bahng, and Y. Tae, “ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed,” *Proceedings of the 29th ACM International Conference on Information and Knowledge Management,* 2020, pp. 1325-1334.
16. Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” *International Conference on Learning Representations*, 2018.
17. C. Li, Y. Zhao, and Z. Zhang, “AMGCN: adaptive multigraph convolutional networks for traffic speed forecasting,” *Applied Intelligence*, Vol. 54, 2024, pp. 2594-2613.
18. L. Bai, L. Yao, C. Li, and X. Wang, “Adaptive graph convolutional recurrent network for traffic forecasting,” *Advances in Neural Information Processing Systems*, Vol. 33, 2020, pp. 17953-17963.
19. G. Yan, H. Li, B. Zhang, et al., “Traffic prediction model considering spatial correlation of non-neighboring nodes,” *Application Research of Computers*, Vol. 42, 2025, pp. 825-833.
20. W. J. Shen, “From roads to graphs: A deep dive into graph neural networks for traffic prediction,” in *Proc. 2025 Int. Conf. Front. Commun. Technol. Cloud Comput. (CFCTCC 2025)*, AIP Conf. Proc., vol. 3435, 2026, p. 030036.
21. C. Daganzo, “The Cell Transmission Model: A Dynamic Representation of Highway Traffic Consistent with the Hydrodynamic Theory,” *Transportation Research Part B: Methodological*, Vol. 28, 1994, pp. 269-287.
22. Z. Cui, K. Henrickson, R. Ke, et al., “Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 21, 2020, pp. 4883-4894.
23. Y. Cai, Z. Zhang, P. Ghamisi, et al., “Fully Linear Graph Convolutional Networks for Semi-Supervised and Unsupervised Classification,” *ACM Transactions on Intelligent Systems and Technology*, Vol. 14, 2023, pp. 1-23.
24. Z. Ye, F. Wu, “Semi-supervised Classification with Adaptive High-Order Graph Embedding,” in *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA), Kunming, China,* 2018, pp. 48-53.
25. J. Li, H. Peng, Y. Cao, et al., “Higher-Order Attribute-Enhancing Heterogeneous Graph Neural Networks,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, 2023, pp. 560-574.



Hui Zhi received the Ph.D. degree in communication and information systems from the Nanjing University of Posts and Telecommunications, in July 2013. She is currently a Researcher with the Key Laboratory of Computational Intelligence and Signal Processing of Ministry of Education, Anhui university, China. She is also an Associate Professor and a Master Tutor with Anhui University. Her current research interests include blockchain, cooperative communication, network coding, and massive MIMO.



Yi-Yuan Feng graduated from Zhengzhou University majoring in

Communication Engineering in 2024. She is currently studying for a master's degree in Electronic Information Engineering at Anhui University, China. Her research interests include RIS-assisted wireless communication, multi-agent assisted communication.



Miao-Miao Duan graduated from Anhui Xinhua University majoring in Internet of Things Engineering in 2021. She is currently studying for a master's degree in Electronic Information Engineering at Anhui University, China. Since 2021, she has been engaged in wireless communication research at the Key Laboratory of Computational Intelligence and Signal Processing of the Ministry of Education. Her research interests include the study of communication access point selection, blockchain, federated learning, and traffic flow prediction methods.