Hierarchical Graph Convolution Networks for Traffic Forecasting

Kan Guo[1,4], Yongli Hu[1*], Yanfeng Sun[1], Sean Qian[2], Junbin Gao[3], Baocai Yin[1,4]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing
[2]Civil and Environmental Engineering and H.John Heinz III College, Carnegie Mellon University, Pittsburgh
[3]Business School, The University of Sydney, Australia
[4]Peng Cheng Laboratory, Shenzhen, 518055, China
guokan@emails.bjut.edu.cn, {huyongli,yfsun,ybc}@bjut.edu.cn, seanqian@cmu.edu, junbin.gao@sydney.edu.au,

Abstract

Traffic forecasting is attracting considerable interest due to its widespread application in intelligent transportation systems. Given the complex and dynamic traffic data, many methods focus on how to establish a spatial-temporal model to express the non-stationary traffic patterns. Recently, the latest Graph Convolution Network (GCN) has been introduced to learn spatial features while the time neural networks are used to learn temporal features. These GCN based methods obtain state-of-the-art performance. However, the current GCN based methods ignore the natural hierarchical structure of traffic systems which is composed of the micro layers of road networks and the macro layers of region networks, in which the nodes are obtained through pooling method and could include some hot traffic regions such as downtown and CBD etc., while the current GCN is only applied on the micro graph of road networks. In this paper, we propose a novel Hierarchical Graph Convolution Networks (HGCN) for traffic forecasting by operating on both the micro and macro traffic graphs. The proposed method is evaluated on two complex city traffic speed datasets. Compared to the latest GCN based methods like Graph WaveNet, the proposed HGCN gets higher traffic forecasting precision with lower computational cost.The website of the code is https://github.com/guokan987/HGCN.git.

1 Introduction

The intelligent transportation system has been a fast growing research field with the development of sensor technology and the diversification of travel modes. One of significant tasks in such systems is how to predict the future traffic state of road network, which has many applications in daily travel such as planning travel routes in advance and guiding the allocation of road usage, etc. Recently, the ride-hailing and ride-sharing services have emerged and been popular in cities. This novel travel mode accumulates a huge volume of traffic data along with the traditional sensor data, which provides abundant traffic data to analyze traffic patterns and achieve traffic forecasting in the complex city road environment.

For traffic forecasting (Ahmed and Cook 1979), it has a long history of development from 1980s. Usually, early researches were based on the historical data of the road network and utilized the simple linear regression methods to predict traffic state in the next few minutes. Then, with the development of statistical technology and regression methods, in order to construct the Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS), some real-time traffic simulation systems were proposed, such as DynaMIT (Ben-Akiva et al. 1998, 2012) and DYNASMART-X (Mahmassani et al. 2005). They integrate the state estimation, traffic assignment and control strategies functions into one system, in which the state estimation adopts Kalman Filters (KFs) and its variants (Wang and Papageorgiou 2005; Tampere and Immers 2007; van Hinsbergen et al. 2012). Different from KFs or its variants, some works focus on the machine learning methods of data-driven models such as Auto Regressive Integrated Moving Average (ARIMA) (Ahmed and Cook 1979; Smith, Williams, and Oswald 2002), Support Vector Regression (SVR) (Wu, Wei, and Su 2004; Cong, Wang, and Li 2016), Random Forest Regression (Leshem and Ritov 2007; Yang and Qian 2018), and so on. For these models, their performances are usually limited by the feature representation capacity.

As the most important component of machine learning, Neural network, especially deep neural network, has powerful representation ability, thus it is widely used in Computer Vision, Natural Language Processing, and Traffic Forecasting (Yu and Chen 1993; Florio and Mussone 1996; Zhou and Nelson 2002). When huge volume of traffic data are accumulated, deep neural network has been used to explore the inner relationship hiding in the traffic data for improving the result of traffic forecasting (Lv et al. 2015). Recurrent Neural Networks (RNNs) such as LSTM (Cui, Ke, and Wang 2016) and the Gated Recurrent Unit (GRU) (Agarap 2017) were also utilized to explore the temporal feature of traffic data. Except for RNNs, Deep Spatial Temporal Convolution Network (DSTCN) (Zhang, Zheng, and Qi 2017) was proposed to learn spatial features with Convolution Neural Network (CNN) and temporal feature with LSTM. However, it needs to transfer the traffic data to image-grid data and doing so destroys the natural connection of road network.

To avoid the disadvantage of CNN, recently, Graph Convolution Network (Bruna et al. 2014; N.Kipf and Welling 2017) was introduced to exploit the non-grid local spatial

*Correspondence Author

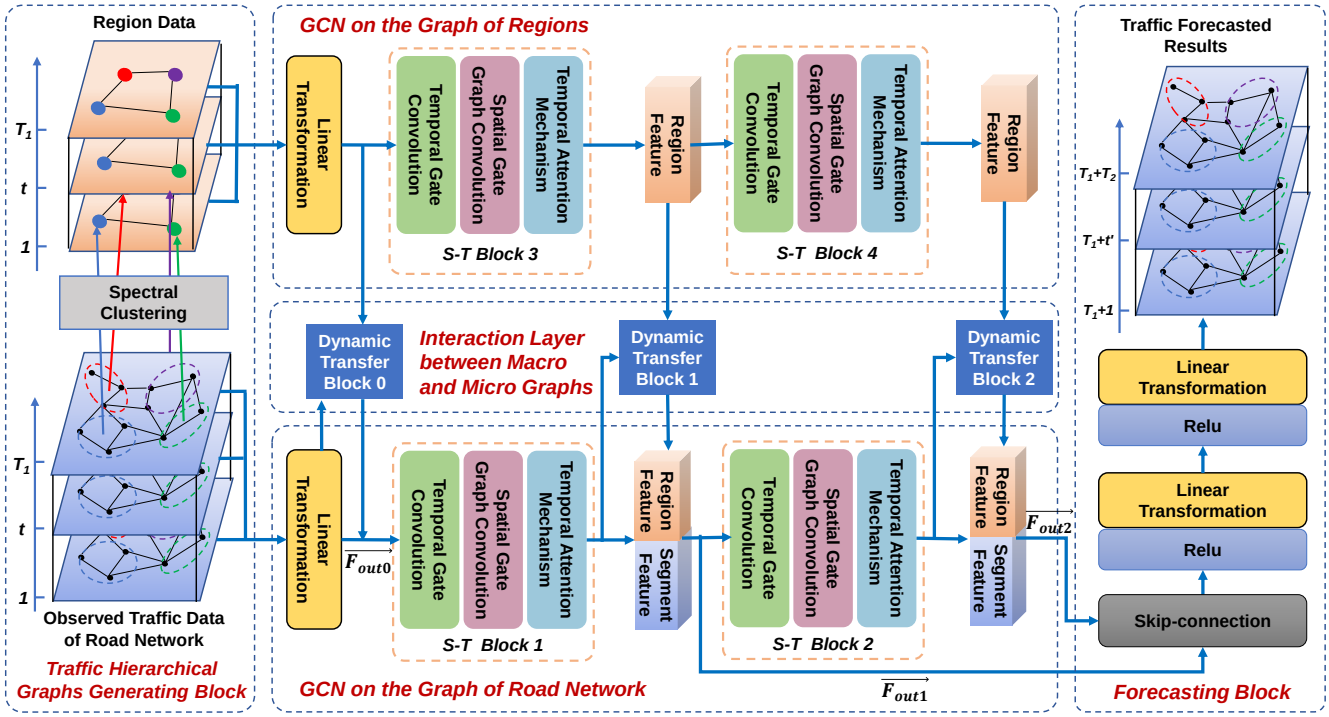


Figure 1: The proposed HGCN for traffic forecasting.

feature through Laplacian matrix. Then Graph Convolution Recurrent Network (GCRN) (Seo et al. 2017) and Gated Spatial-Temporal Graph Convolution Network (Gated-STGCN) (Yu, Yin, and Zhu 2018) were proposed to solve the sequence modeling problem of traffic forecasting by combining GCN with RNNs or Gated-CNN. However, they are based on the empirical or handcrafted Laplacian matrix, which ignores many useful spatial relations hiding in the traffic data. Thus, the data-driven parametric Laplacian matrix was proposed, such as Optimized Graph Convolution Recurrent Neural Network (OGCRNN) (Guo et al. 2020a) and Optimized Temporal-Spatial Gated Graph Convolution Network (OTSGGCN) (Guo et al. 2020b), in which the same size of parametric matrix was added on Laplacian matrix to reveal more relations between nodes. Besides the parametric Laplacian matrix, the graph WaveNet (GWNET) (Wu et al. 2019) was proposed based on diffusion GCN (Li et al. 2018) to deal with a directed graph in the traffic forecasting.

As the GCN based methods are more suitable to process the irregular graph type of traffic data, they have achieved state-of-the-art traffic forecasting performance. In spite of this, the current GCN based methods generally deal with the traffic data based on the fundamental road network, the basic and single-layer of graph of the traffic system, in which the nodes represent road segments and the links mean pairs of correlative road segments. However, these methods ignore the natural hierarchical structure of traffic system, which not only includes the basic micro layer of road network but also has the macro layers of networks with the nodes representing the hot traffic regions covering downtowns, CBDs or other important blocks, etc. The feature and information of region or community play an important role in traditional transformation theory for traffic planning or macro feature analysis (Chen et al. 2006), while the current GCN based traffic forecasting methods have seldom utilized this information at present.

From the above analysis, in this paper, we propose a novel Hierarchical Graph Convolution Networks (HGCN) for traffic forecasting. As shown in Figure 1, the proposed HGCN is featured in multi-layers of GCN for traffic forecasting, i.e. the micro layer of road network and the macro layer of network with region nodes, which are constructed from the clustering of the nodes of road segments. Additionally, the most important contribution of this work is that we construct the interaction between the micro and macro layers of GCNs, which integrates the different scales of features of road segments and regions for improving the traffic forecasting performance. The main contributions of our paper are summarized as follows,

- A novel Hierarchical Graph Convolution Network based on pooling is proposed for traffic forecasting considering both the road segments and regions feature of traffic system;
- The multiple GCNs on different layers of traffic graphs are properly integrated by introducing interaction of dynamic transfer blocks;
- The proposed method is tested on two traffic datasets captured from two big cities and the results outperform the state-of-the-art related works.

2 Related Work

Traffic Forecasting

Traffic forecasting can be modeled as a function which maps the observed traffic data of the road network to the future traffic state such as flow, speed, etc. From another perspective, traffic forecasting is also a sequence modeling problem which constructs the relation between the historical data and future data through the moving window technique. So, for the traffic data in the ith road segment at t time, we define it as $x_t^i \in \mathbb{R}^D$, in which D is the feature size of x_t^i. As a result, the traffic data of the total road network at time t can be defined as $X_t = [x_t^1, \ldots, x_t^i, \ldots, x_t^N]^T \in \mathbb{R}^{N \times D}$, where N is the number of road segments. In the traffic forecasting, we forecast the traffic state for the next T_2 times by using T_1 historical data, so the input sequence of historical data can be defined as $\{X_1, \ldots, X_t, \ldots, X_{T_1}\} \in \mathbb{R}^{N \times T_1 \times D}$, and its forecasting sequence of the future can be defined as $\{\hat{X}_{(T_1+1)}, \ldots, \hat{X}_{(T_1+T_2)}\} \in \mathbb{R}^{N \times T_2 \times D}$, and its ground truth is denoted by $\{X_{(T_1+1)}, \ldots, X_{(T_1+T_2)}\} \in \mathbb{R}^{N \times T_2 \times D}$. To represent the natural topological connection of road network, the graph of road network is defined as $G = (V, E, A)$, where $V \in \mathbb{R}^N$ is the road segment(node) set, E is the edge set, $A \in \mathbb{R}^{N \times N}$ is the adjacent matrix of G.

Graph Convolution Network

Graph Convolution Network (Bruna et al. 2014) is derived from Graph Spectral Theory (Chung 1992). Different from CNN, GCN specially deals with irregular graph data depending on the decomposition of graph Laplacian matrix, which achieves the process of filter in the frequency domain as follows,

$$g_\theta \star G(x) = g_\theta(L)x = U g_\theta(\Lambda) U^T x \tag{1}$$

where $L = U\Lambda U^T$ is the graph Laplacian matrix, and U is the Fourier basis of G and $\Lambda = diag([\lambda_1, \ldots, \lambda_N]) \in \mathbb{R}^{N \times N}$. So, the original GCN is dependent on the decomposition of L, which is a computationally high demanded process. For this purpose, the fast GCN (Defferrard, Bresson, and Vandergheynst 2016) was proposed to solve this computational problem as follows,

$$g_\theta \star G(x) = g_\theta(L)x = \sum_{m=0}^{M-1} \theta_m C_m(\tilde{L}) x \tag{2}$$

where θ_m is the learnable parameters and $m = 0, \ldots M-1$ is the order of the Chebyshev Polynomials $C_m(\tilde{L}) = 2\tilde{L}C_{m-1}(\tilde{L}) - C_{m-2}(\tilde{L})$ and $C_1(\tilde{L}) = \tilde{L}, C_0(\tilde{L}) = I_N$. $\tilde{L} = \frac{2}{\lambda_{max}} L - I_N$ is scaled Laplacian matrix for better representation capacity.

To process a graph with two different directions: the node of input and output, the diffusion GCN (Li et al. 2018) was proposed based on the spatial GCN (Duvenaud et al. 2015) as follows,

$$g_\theta \star G(x)^d = \sum_{m=0}^{M-1} \theta_{m,f} P_f^m x + \theta_{m,b} P_b^m x \tag{3}$$

where $P_f = A/rowsum(A)$, and P_f^m is the m-order matrix power of P_f. $P_b = A^T/rowsum(A^T)$, and P_b^m is the m-order matrix power of P_b. Then $\theta_{m,f}$ and $\theta_{m,b}$ are parameters. Except for P_f and P_b, GWNET (Wu et al. 2019) added the optimized Laplacian matrix to learn the parameterized spatial relation from data, i.e. it is defined as $\tilde{A}_{adp} = Softmax(Relu(E_1 E_2^T))$, where $E_1 \in \mathbb{R}^{N \times E}$ and $E_2 \in \mathbb{R}^{N \times E}$ are two parameters in the size of E. Thus, GWNET can be represented as follows,

$$g_\theta \star G(x)^{adp} = \sum_{m=0}^{M-1} \theta_{m,f} P_f^m x + \theta_{m,b} P_b^m x + \theta_{m,adp} \tilde{A}_{adp}^m x \tag{4}$$

where $\tilde{A}_{adp}^m$ is the m-order matrix power of $\tilde{A}_{adp}$.

3 Methodology

The framework of HGCN is shown in Figure 1. It contains five components which are labeled in red text. We will describe each component in details in the sequel.

Traffic Hierarchical Graphs Generating by Spectral Clustering

The micro graph of a road network can be constructed from its natural structure. In this paper, we use the distance between the nodes of road segments to construct the graph of road network, i.e. we use the GPS coordinates of the terminal point of road segments to calculate their distance and omit these segments with distance great than a given threshold. The value of the node is the observed traffic data like the speed, flow or density on the road segment. We denote the observed traffic data of the graph of road network by $\overrightarrow{X} = \{X_1, \ldots, X_t, \ldots, X_{T_1}\} \in \mathbb{R}^{N \times T_1 \times D}$, which will be used as the input for the following GCN.

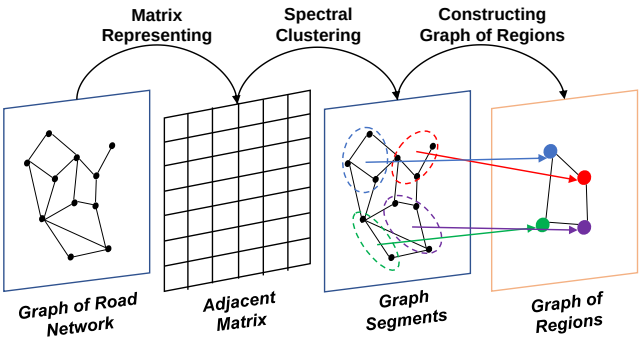


Figure 2: The sketch map of the region data generation.

For constructing the macro graph of regions from road network, the ideal way is to use the actual traffic hot regions or communities as the nodes. However, how to detect and recognize these hot regions in a complex traffic system is another challenging issue. For simplicity, we utilize the spectral clustering method on the graph of road network to construct the macro graph of regions. As shown in Figure 2, we first obtain the adjacent matrix from the

graph of road network. Then we conduct spectral clustering on the Laplacian matrix of the adjacent matrix, and get a partition of the road network. Each cluster of the partition can be regarded as a macro node of the graph of regions. For example, in Figure 2, the road network is clustered into four regions. The value of the macro node is the combination of the mean and minimum values of the micro nodes in the cluster. The edge of the macro graph is constructed based on the micro graph, for example, if $i \in region1$ and $j \in region2$, and the ith node connects to jth node in micro graph, then $region1$ connects to $region2$ in macro graph. From this construction, we denote the macro graph of regions by $\overrightarrow{X^R} = \{X_1^R, \ldots, X_t^R, \ldots, X_{T_1}^R\} \in \mathbb{R}^{N^R \times T_1 \times D}$, where N^R is the number of macro traffic regions.

GCN on the Graph of Regions

As shown in Figure 1, GCN on the graph of regions is implemented by one Linear Transformation and two stacked Spatial-Temporal Blocks (S-T Blocks). The former realizes the transformation for the input region data $\overrightarrow{X^R}$, and the latter is designed to extract the spatial and temporal feature of the region data by GCN.

Spatial-Temporal Block The S-T Block consists of three parts: Temporal Gate Convolution, Spatial Gate Graph Convolution and Temporal Attention Mechanism, which aim to learn the local temporal feature, the local spatial-temporal feature and the global temporal relation of region data, respectively. The details are given in following.

(1) Temporal Gate Convolution: It is presented to parallelly extract hierarchical local temporal features by using gate convolution in timeline. Compared to LSTM, it is highly efficient for the long-range time series data. Specifically, it can be represented as follows,

$$\begin{aligned} TC(\overrightarrow{X}) &= \Phi \star \overrightarrow{X} = Conv_{t_s}(\overrightarrow{X}) \\ (\vec{\beta}_1, \vec{\beta}_2) &= split(TC(\overrightarrow{X})) \\ TGC(\overrightarrow{X}) &= tanh(\vec{\beta}_1) * sigmoid(\vec{\beta}_2) \end{aligned} \tag{5}$$

where $Conv_{t_s}$ represents the temporal Convolution operator in time size and its kernel size is t_s, and $split$ represents the operator of equally partition, thus $TGC(\overrightarrow{X}) \in \mathbb{R}^{N^R \times T_1^1 \times D_1}, \vec{\beta}_1 \in \mathbb{R}^{N^R \times T_1^1 \times D_1/2}, \vec{\beta}_2 \in \mathbb{R}^{N^R \times T_1^1 \times D_1/2}$, $T_1^1 = T_1 - (2 * t_s - 2)$, and $sigmoid, tanh$ are activation functions.

To expand receptive filed of gate convolution, we adopt the dilation convolution (Wu et al. 2019) to replace original convolution with the dilation coefficient $dil = 2$. So, the above Temporal Gate Convolution can be revised as follows,

$$\begin{aligned} TC(\overrightarrow{X}) &= \Phi \star \overrightarrow{X} = Conv_{t_s}^{dil}(\overrightarrow{X}) \\ (\vec{\beta}_1, \vec{\beta}_2) &= split(TC(\overrightarrow{X})) \\ TGC(\overrightarrow{X}) &= tanh(\vec{\beta}_1) * sigmoid(\vec{\beta}_2) \end{aligned} \tag{6}$$

(2) Spatial Gate Graph Convolution: To explore the spatial feature and the local temporal feature simultaneously, we propose to embed the diffusion graph convolution in (4) into the Temporal Gate Convolution (without dilation convolution) as follows,

$$\begin{aligned} GTC(\overrightarrow{X}) =& \sum_{m=0}^{M-1} \Phi_{m,f} \star P_f^m x + \Phi_{m,b} \star P_b^m x \\ &+ \Phi_{m,adp} \star \tilde{A}_{adp}^m x \\ (\vec{\beta}_1, \vec{\beta}_2) =& split(GTC(\overrightarrow{X})) \\ DGGC(\overrightarrow{X}) =& tanh(\vec{\beta}_1) * sigmoid(\vec{\beta}_2) \end{aligned} \tag{7}$$

where $DGGC(\overrightarrow{X}) \in \mathbb{R}^{N^R \times T_1^2 \times D_1}, T_1^2 = T_1^1 - (t_s - 1) = T_1 - (3 * t_s - 3)$, $\tilde{A}_{adp} = norm(Relu(E_1 E_2^T))$ and $norm$ is defined as follows,

$$\begin{aligned} A_{adp} &= Relu(E_1, E_2^T) \\ D_{adp_{ii}} &= \sum_j A_{adp_{ij}} \\ D_{adp1} &= diag(1/(D_{adp_{ii}})) \\ \tilde{A}_{adp} &= D_{adp1} A_{adp} \end{aligned} \tag{8}$$

Here, we use $norm$ instead of $softmax$ in (4) to avoid full connection of all nodes and keep the sparsity of $\tilde{A}_{adp}$.

(3) Temporal Attention Mechanism: To further explore the global temporal relation, we utilize the Temporal Attention (Feng et al. 2017) to capture the large scale temporal correlation of traffic data $\overrightarrow{X}$ as follows,

$$\begin{aligned} E =& V_e \sigma((\overrightarrow{X})^T U_1) U_2 ((\overrightarrow{X}) U_3)^T + b_e) \\ E'_{i,j} =& \frac{exp(E_{i,j})}{\sum_{j=1}^{T_1^2} exp(E_{i,j})} \\ Tatt(\overrightarrow{X}) =& E' \overrightarrow{X} \end{aligned} \tag{9}$$

where $V_e, b_e \in \mathbb{R}^{T_1^2 \times T_1^2}$, $U_1 \in \mathbb{R}^{N \times 1}$, $U_2 \in \mathbb{R}^{D_1 \times N}$, $U_3 \in \mathbb{R}^{D_1 \times 1}$. From the above, we summary S-T Block as Algorithm 1.

Algorithm 1 Spatial-Temporal Block

Require:
The current input traffic data $\overrightarrow{X} = \{X_1, \ldots, X_t, \ldots, X_{T_1}\} \in \mathbb{R}^{N \times T_1 \times D}$; The output feature $\overrightarrow{F3} \in \mathbb{R}^{N \times T_1^2 \times D_1}$

1: $\overrightarrow{F} = TGC(\overrightarrow{X})$;
2: $\overrightarrow{F_1} = DGGC(\overrightarrow{F})$;
3: $\overrightarrow{F_2} = Tatt(\overrightarrow{F_1})$;
4: $\overrightarrow{F_3} = Batch\_norm(\overrightarrow{F_2} + conv_{1\times1}(\overrightarrow{X})[:, -T_1^2 :, :])$;
5: **return** $\overrightarrow{F_3}$

GCN on the Graph of Road Network

As shown in Figure 1, GCN on the Graph of Road Network shares the same structure to GCN on the Graph of Regions. Except for the different input data, i.e. $\overrightarrow{X}$ instead of $\overrightarrow{X^R}$,

the main difference is that the output feature of S-T Block in the micro graph is combined with the feature of the macro graph. Thus, the graph convolution on the road network will be affected by the convolution of region graph, and it is the unique feature of our method, which utilizes the natural hierarchical structure of the transportation system and its observed data for improving traffic forecasting. In the following subsection, we will describe the interaction layer between the macro and micro graphs in detail.

Interaction Layer between Macro and Micro Graphs

To realize the interaction between the macro and micro graph convolutions, we propose a dynamic transfer block to fuse the region features and road segment features. First, we construct a transfer function to form the combined feature, in which if the road node i belongs to the region j, we copy the region j's feature and concatenate it with the feature of the road segment i, i.e. we define a transformation matrix $Tran \in \mathbb{R}^{N \times N^R}$ as follows,

$$[Tran]_{ij} = \begin{cases} 1, \text{if the node } i \text{ belongs to the region } j; \\ 0, \text{ else.} \end{cases}$$

Then, for the road segment feature $\overrightarrow{F} \in \mathbb{R}^{N \times T_1^2 \times D_1}$ and the Region feature $\overrightarrow{F^R} \in \mathbb{R}^{N^R \times T_1^2 \times D_1}$, the feature transfer function can be formulated as follows,

$$\begin{aligned} \overrightarrow{F^R_{Tran}} &= Tran * \overrightarrow{F^R} \\ \overrightarrow{F_{out}} &= Concat(\overrightarrow{F}, \overrightarrow{F^R_{Tran}}) \end{aligned} \tag{10}$$

where $\overrightarrow{F^R_{Tran}} \in \mathbb{R}^{N \times T_1^2 \times D_1}$ and the output feature of transfer $\overrightarrow{F_{out}} \in \mathbb{R}^{N \times T_1^2 \times (2*D_1)}$.

As the dynamic change characteristic of traffic data, the relation between road segment and region is also changeable. So we further propose a dynamic transfer matrix $Tran^d \in \mathbb{R}^{N \times N^R}$ based on the attention mechanism similar to (9) with the following form.

$$\begin{aligned} E^d &= \sigma((\overrightarrow{F})^T U_1) U_2 ((\overrightarrow{F^R}) U_3)^T + b_e) \\ E^d &= E^d - mean(E^d, axis = 0) \\ Tran^d &= \sigma(E^d) * Tran \end{aligned} \tag{11}$$

From this, Dynamic Transfer Block can be finally defined as follows,

$$\begin{aligned} \overrightarrow{F^R_{Tran}} &= Tran^d * \overrightarrow{F^R} \\ \overrightarrow{F_{out}} &= Concat(\overrightarrow{F}, \overrightarrow{F^R_{Tran}}) \end{aligned} \tag{12}$$

Traffic Forecasting Block

We utilize the fused feature of two graphs for traffic forecasting. To get more information from different stage features, we design a skip-connection to process these features. Then, we integrate the output of skip-connection and feed it into the predicting block for forecasting results. The predicting block is composed of two stacked layers of $relu$ with linear transformation, as shown in Figure 1. The procedure of the forecasting block can be formulated as follows,

$$\begin{aligned} \overrightarrow{F_{skip1}} &= \overrightarrow{F_{out1}} S_1, \overrightarrow{F_{out1}} \in \mathbb{R}^{N \times T_1^2 \times (2*D_1)} \\ \overrightarrow{F_{skip2}} &= \overrightarrow{F_{out2}} S_2, \overrightarrow{F_{out2}} \in \mathbb{R}^{N \times t_w \times (2*D_1)} \\ \overrightarrow{F_{sum}} &= Relu(\overrightarrow{F_{skip1}}[:, -t_w:, :] + \overrightarrow{F_{skip2}}) \\ \overrightarrow{F_{sum1}} &= Relu(\overrightarrow{F_{sum}} W_1) \\ Output &= \overrightarrow{F_{sum1}} W_2 \end{aligned} \tag{13}$$

where $S_1 \in \mathbb{R}^{(2*D_1) \times D_2}, S_2 \in \mathbb{R}^{(2*D_1) \times D_2}, W_1 \in \mathbb{R}^{t_w \times 1 \times D_2 \times D_3}, W_2 \in \mathbb{R}^{D_3 \times T_2}$ are parameters and $Output = \{\hat{X}_{(T_1+1)}, \ldots, \hat{X}_{(T_1+T_2)}\}$. W_1 will decrease the time size of $\overrightarrow{F_{sum}}$ to 1.

In this paper, we use the Mean Absolute Error (MAE) to form the loss function, i.e. for the ground truth $Truth = \{X_{(T_1+1)}, \ldots, X_{(T_1+T_2)}\}$, the loss function can be represented as follows,

$$\begin{aligned} loss &= MAE(Output, Truth) \\ &= \frac{\sum_{i=1}^{T_2} \sum_{j=1}^{N} |(\hat{X}^j{}_{(T_1+i)} - X^j_{(T_1+i)})|}{T_2 * N} \end{aligned} \tag{14}$$

Finally we summarize the proposed HGCN as shown in Algorithm 2.

Algorithm 2 The HGCN algorithm for traffic forecasting.

Require:

The observed traffic data $\overrightarrow{X} = \{X_1, \ldots, X_t, \ldots, X_{T_1}\} \in \mathbb{R}^{N \times T_1 \times D}$;

1: Generating the region data $\overrightarrow{X^R} = \{X^R_1, \ldots, X^R_t, \ldots, X^R_{T_1}\} \in \mathbb{R}^{N^R \times T_1 \times D}$ from $\overrightarrow{X}$;
2: Get the input feature of segment and region $\overrightarrow{F_0}$ and $\overrightarrow{F_0^R}$ from $\overrightarrow{X}$ and $\overrightarrow{X^R}$ by Linear Transformation;
3: Get the combined feature $\overrightarrow{F_{out0}}$ from $\overrightarrow{F_0}, \overrightarrow{F_0^R}$ by Dynamic Transfer Block 0;
4: Get the segment feature $\overrightarrow{F_1}$ from $\overrightarrow{F_{out0}}$ by S-T Block 1;
5: Get the region feature $\overrightarrow{F_3^R}$ from $\overrightarrow{F_0^R}$ by S-T Block 3;
6: Get the combined feature $\overrightarrow{F_{out1}}$ from $\overrightarrow{F_1}, \overrightarrow{F_3^R}$ by Dynamic Transfer Block 1;
7: Get the segment feature $\overrightarrow{F_2}$ from $\overrightarrow{F_{out1}}$ by S-T Block 2;
8: Get the region feature $\overrightarrow{F_4^R}$ from $\overrightarrow{F_3^R}$ by S-T Block 4;
9: Get the combined feature $\overrightarrow{F_{out2}}$ from $\overrightarrow{F_2}, \overrightarrow{F_4^R}$ by Dynamic Transfer Block 2;
10: Get the $Output$ of HGCN from $\overrightarrow{F_{out1}}, \overrightarrow{F_{out2}}$ by (13);
11: **return** $Output$
12: Calculate the $loss$ of HGCN by (14).

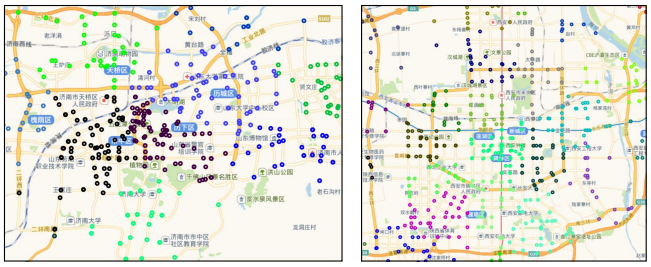
4 Experiments

In our experiments, we evaluate the proposed HGCN on two real-world traffic datasets to compare with state-of-the-art related methods.

Data	Method	30 min			1 hour			2 hour		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
JiNan	HA	5.69	7.60	20.02%	5.69	7.60	20.02%	5.69	7.60	20.02%
	ARIMA	3.96	6.14	14.12%	4.48	6.56	16.10%	5.09	7.01	18.24%
	LSTM	3.21	4.85	12.85%	3.67	5.48	14.92%	4.30	6.26	17.38%
	GRU	3.20	4.85	12.82%	3.67	5.47	14.86%	4.30	6.26	17.41%
	GCRN	2.99	4.54	12.11%	3.30	4.96	13.53%	3.71	5.45	15.13%
	Gated-STGCN	2.96	4.48	11.85%	3.29	4.92	13.33%	3.71	5.44	15.03%
	OGCRNN	3.12	4.60	12.35%	3.39	4.93	13.40%	3.67	5.24	14.38%
	OTSGGCN	3.00	4.48	11.86%	3.26	4.73	13.11%	3.57	5.21	14.32%
	GWNET	**2.89**	4.37	11.49%	3.16	4.71	12.52%	3.52	5.16	13.77%
	HGCN_WH(our)	**2.89**	4.38	11.59%	3.15	4.73	12.70%	3.43	5.04	13.65%
	HGCN_WDF(our)	2.91	4.45	11.58%	3.15	4.78	12.57%	3.41	5.10	13.45%
	HGCN(our)	**2.89**	**4.37**	**11.35%**	**3.11**	**4.68**	**12.31%**	**3.36**	**5.02**	**13.31%**
XiAn	HA	6.02	8.16	21.79%	6.02	8.16	21.79%	6.02	8.16	21.79%
	ARIMA	3.70	6.05	12.96%	4.26	6.57	15.28%	5.04	7.24	18.26%
	LSTM	3.16	4.83	11.92%	3.70	5.52	14.22%	4.52	6.53	17.42%
	GRU	3.15	4.82	11.96%	3.69	5.52	14.25%	4.51	6.53	17.50%
	GCRN	2.92	4.48	11.13%	3.28	4.98	12.83%	3.78	5.60	14.91%
	Gated-STGCN	2.89	4.48	11.09%	3.23	4.94	12.69%	3.73	5.52	14.56%
	OGCRNN	2.94	4.47	11.26%	3.20	4.79	12.43%	3.54	5.19	13.73%
	OTSGGCN	2.87	4.42	11.14%	3.16	4.79	12.48%	3.50	5.20	13.80%
	GWNET	2.76	4.26	10.46%	3.03	4.61	11.71%	3.44	5.10	13.22%
	HGCN_WH(our)	**2.75**	4.28	**10.40%**	3.00	4.61	11.55%	3.34	5.05	13.17%
	HGCN_WDF(our)	2.77	4.33	10.50%	3.02	4.66	11.69%	3.30	4.99	12.72%
	HGCN(our)	**2.75**	**4.26**	10.46%	**2.96**	**4.54**	**11.44%**	**3.24**	**4.85**	**12.52%**

Table 1: The traffic forecasting results of different methods on JiNan and XiAn datasets.

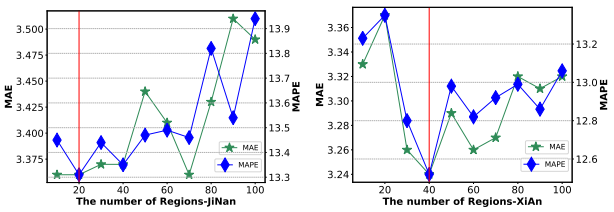
Experimental Settings



(a) JiNan City (b) XiAn City

Figure 3: The part of road segments shown on the maps. The points are the terminal points of the road segments with the different color representing its traffic region belonged to.

Datasets Two traffic speed datasets used in our experiments are collected by Didi Chuxing GAIA Initiative (https://gaia.didichuxing.com) in JiNan and XiAn cities in China, as Figure 3 shown. These datasets contain the average speed of road segments in one with sampling rate of one sample per 10 minutes. The total sample number of the two datasets is 52286 each. In the two datasets, there are 561 and 792 road segments (nodes) in city center area for JiNan and XiAn, respectively. We adopt Z-score normalization to process the data in both datasets. Based on a thresholded Gaussian kernel (Shuman et al. 2013), we construct the adjacent matrix of the road network. Each dataset is splitted into 60% for training, 20% for validation and 20% for test with chronological order. We train models in the training-set, and according to the results of the validation-set choose the optimal parameters to test the model on the test-set. It is noted that missing values are excluded in both cases from training-set, validation-set and test-set.



(a) (b)

Figure 4: MAE and MAPE of 2 hours forecasting with different N^R on the two datasets.

Comparison methods The proposed method is compared with nine traffic forecasting methods: Historical Average(HA), ARIMA, LSTM(Cui, Ke, and Wang 2016), GRU(Agarap 2017), GCRN(Seo et al. 2017), Gated-STGCN(Yu, Yin, and Zhu 2018), OGCRNN(Guo et al. 2020a) , OTSGGCN(Guo et al. 2020b) , GWNET(Wu et al. 2019). To estimate the impact of GCN on the graph of regions, we make a version of our model without hierarchical structure, i.e. only using road segment feature, denoted by HGCN_WH. To further evaluate the efficiency of the proposed Dynamic Transfer Block in HGCN, we replace $Tran^d$ with $Tran$ resulting in a model without a dynamic transfer matrix, denoted by HGCN_WDF. The performance of all methods is measured by three metrics: Mean Absolute

Error(MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE).

Parameters Setting In our experiments, we set the order of GCN's order $M = 3$ in the S-T Block according to the previous works (N.Kipf and Welling 2017; Wu et al. 2019). The size of the temporal kernel $t_s = 3$ in S-T Block 1 and S-T Block 3, and $t_s = 2$ in S-T Block 2 and S-T Block 4. The forecasting time interval and input time interval is equal, i.e. $T_1 = 12, T_2 = 12$, thus $t_w = 3$. The feature sizes are $D = 1, D_1 = 32, D_2 = 256, D_3 = 512$. The size of initialize node embedding in $\tilde{A}_{adp}$ is 10, i.e. $E = 10$. To select optimal setting of the number of regions N^R, we let $N^R = 20$ for JiNan dataset and $N^R = 40$ for XiAn dataset according to the results on the validation-set, which are shown in Figure 4. The proposed model is implemented by Pytorch 1.2.0 on a virtual workstation with a 11G memory Nvidia RTX 2080Ti. The batch size is 64. The Adam Optimization is utilized. The original learning rate is 0.001. We train 50 epochs in the training phase.

Experimental Results

Traffic Forecasting results The 30 minutes, 1 hour and 2 hours traffic forecasting results of different methods on the two datasets are shown in Table 1. It is shown that the proposed HGCN has the best performance compared with other methods in all metrics, including GWNET which is the latest related GCN based method with state-of-the-art performance. Overall, the GCN based methods, including GCRN, Gated-STGCN, OGCRNN, OTSGGCN, GWNET and our HGCN, are better than the remaining methods, which means that the graph convolution is more suitable to process the traffic data with graph structure. Additionally, the deep learning based methods are better than traditional HA and ARIMA methods, which validates that deep learning methods have stronger ability to learn useful features for this application.

Effect of the Hierarchical Structure To estimate the effect of hierarchical structure and dynamic transfer block, we design the validation experiments and show the forecasting results of HGCN_WH and HGCN_WDF in Table 1. From the results, we can conclude that both of hierarchical structure and dynamic transfer block of HGCN are effective as the results of HGCN outperform both of HGCN_WH and HGCN_WDF.

From the maps in Figure 3, one can further observe the effect of the hierarchical structure intuitively. It is shown that the road segments are divided into different regions which cover city hot regions such as school, supermarket, airport, etc. Thus the hierarchical structure meets the natural property of the city transportation system. Additionally, our region partition is more detailed than the real city administrative division. For example, XiAn City has 4 real communities in the center of the city, which is too rough compared with the optimal 40 region division in Figure 4(b).

As for the effect of the Dynamic Transfer Block, we plot the dynamic transfer matrix $Tran^d \in \mathbb{R}^{N \times N^R}$ as a heatmap in Figure 5. It is shown that the values of the matrix change

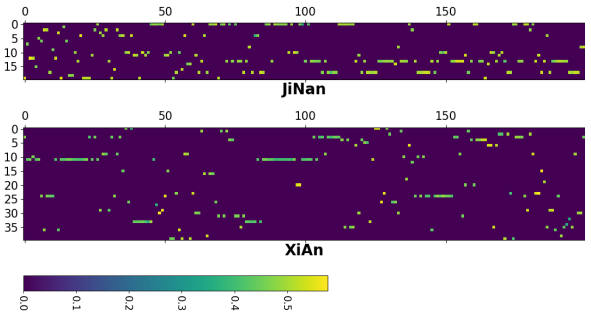


Figure 5: The vision of the part of dynamic transfer matrix on the two datasets in the one time step.

from 0.3 to 0.6, which means the Dynamic Transfer Block successfully controls the information flow from the region graph to the road segment graph.

Model	Computation Time	
	Training(s/epoch)	Inference(s)
OGCRNN	27.65/39.28	4.61/5.96
OTSGGCN	30.27/46.80	3.13/4.84
GWNET	165.73/235.95	12.01/25.35
HGCN_WH	55.38/81.02	4.96/7.61
HGCN_WDF	51.07/101.95	5.13/8.61
HGCN	74.03/106.52	5.98/8.89

Table 2: The computation time on the JiNan and XiAn dataset.

Computation Time We compare the computation time of OGCRNN, OTSGGCN and GWNET with HGCN_WH, HGCN_WDF and HGCN on the XiAn dataset. The results are shown in Table 2. It indicates that although HGCN is slower than OGCRNN and OTSGGCN, it has accuracy far better than these methods. HGCN is slightly slower than HGCN_WH and HGCN_WDF, which illustrates that the cost of hierarchical structure and dynamic transfer block is deserved. Compared with the latest GWNET method, HGCN is not only two times faster than it but also has accuracy higher than GWNET, which validates that HGCN is an efficient method with high performance.

5 Conclusion

In this paper, a novel graph convolution network, namely HGCN, was proposed to forecast traffic data. Different from the current GCN based methods, which only use road segments information in graph convolution, HGCN constructs a two-stream graph network to consider micro and macro traffic information. The proposed method is evaluated on two real-world traffic datasets. The experimental results show that the proposed method outperforms the related state-of-the-art traffic forecasting methods. However, considering the dynamic complexity of the road network and the interference of weather or other factors, the more data sources should be introduced in the traffic forecasting and their GCN framework is worth exploring in future work.

Acknowledgment

The research project is supported by National Natural Science Foundation of China under Grant (No. U19B2039, 61632006, 61672071, U1811463, 61772048, 61806014, 61906011, 61902053), Beijing Natural Science Foundation (No. 4184082, 4204086), Beijing Talents Project (No.2017A24), Beijing Outstanding Young Scientists Projects (BJJWZYJH01201910005018), and Data source: Didi Chuxing GAIA Initiative.

References

Agarap, A. F. 2017. A Neural Network Architecture Combining Gated Recurrent Unit(GRU) and Support Vector Machine(SVM) for Intrusion Detection in Network Traffic data. In *arXiv: 1709.03082*.

Ahmed, M. S.; and Cook, A. R. 1979. Analysis of freeway traffic time-series data by using BoxJenkins techniques. *Transportation Research Record* 722: 1–9.

Ben-Akiva, M.; Bierlaire, M.; Koutsopoulos, H.; and Mishalani, R. 1998. DynaMIT: a simulation-based system for traffic prediction. In *DACCORD Short Term Forecasting Workshop*. Delft The Netherlands.

Ben-Akiva, M. E.; Gao, S.; Wei, Z.; and Wen, Y. 2012. A Dynamic Traffic Assignment Model for Highly Congested Urban Networks. *Transportation Research Part C: Emerging Technologies* 24: 62–82.

Bruna, J.; Zaremba, W.; Szalm, A.; and LeCun, Y. 2014. Spectral Networks and Deep Locally Connected Networks on Graphs. In *International Conference on Learning Representations(ICLR)*.

Chen, Y.; Zhang, Y.; Hu, J.; and Yao, D. 2006. Pattern Discovering of Regional Traffic Status with Self-Organizing Maps. In *2006 IEEE Intelligent Transportation Systems Conference*, 647–652.

Chung, F. 1992. *Spectral Graph Theory*. American Mathematical Society.

Cong, Y.; Wang, J.; and Li, X. 2016. Traffic flow forecasting by a least squares support vector machine with a fruit fly optimization algorithm. *Procedia Engineering* 137: 59–68.

Cui, Z.; Ke, R.; and Wang, Y. 2016. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In *6th International Workshop on Urban Computing*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolution Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems(NIPS)*.

Duvenaud, D.; Maclaurin, D.; AguileraIparraguirre, J.; Bombarelli, R. G.; Hirzel, T.; an AspuruGuzik, A.; and Adams, R. P. 2015. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Advances in Neural Information Processing Systems(NIPS)*, 2224–2232.

Feng, X.; Guo, J.; Qin, B.; Liu, T.; and Liu, Y. 2017. Effective Deep Memory Networks for Distant Supervised Relation Extraction. In *International Joint Conference on Artificial Intelligence(IJCAI)*, 4002–4008.

Florio, L.; and Mussone, L. 1996. Neural-network models for classification and forecasting of freeway traffic flow stability. *Control Engineering Practice* 4(2): 153–164.

Guo, K.; Hu, Y.; Qian, Z. S.; Liu, H.; Zhang, K.; Sun, Y.; Gao, J.; and Yin, B. 2020a. Optimized Graph Convolution Recurrent Neural Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems* 1–12.

Guo, K.; Hu, Y.; Qian, Z. S.; Sun, Y.; Gao, J.; and Yin, B. 2020b. An Optimized Temporal-Spatial Gated Graph Convolution Network for Traffic Forecasting. *IEEE Intelligent Transportation Systems Magazine* .

Leshem, G.; and Ritov, Y. 2007. Traffic flow prediction using adaboost algorithm with random forests as a weak learner. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 19, 193–198.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations(ICLR)*.

Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; and Wang, F. 2015. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems* 16: 865–873.

Mahmassani, H. S.; Fei, X.; Eisenman, S.; Zhou, X.; and Qin, X. 2005. *Dynasmart-x evaluation for real-time TMC application: chart test bed*. Maryland Transportation Initiative, University of Maryland, College Park, Maryland.

N.Kipf, T.; and Welling, M. 2017. SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTION NETWORKS. In *International Conference on Learning Representations(ICLR)*.

Seo, Y.; Defferrard, M.; Vandergheynst, P.; and Bresson, X. 2017. Structured Sequence Modeling with Graph Convolutional Recurrent Networks. In *International Conference on Learning Representations(ICLR)*.

Shuman, D. I.; Narang, S. K.; Frossard, P.; Ortega, A.; and Vandergheynst, P. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* 30(3): 83–98.

Smith, B. L.; Williams, B. M.; and Oswald, R. K. 2002. Camparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies* 10(4): 303–321.

Tampere, C. M.; and Immers, L. H. 2007. An Extended Kalman Filter Application for Traffic State Estimation Using CTM with Implicit Mode Switching and Dynamic Parameters. In *Proceedings of the 2007 IEEE Intelligent Transportation Systems Conference*. IEEE.

van Hinsbergen, C. P. I. J.; Schreiter, T.; Zuurbier, F. S.; van Lint, J. W. C. H.; and van Zuylen, H. J. 2012. Localized Extended Kalman Filter for Scalable Real-Time Traffic State Estimation. *IEEE Transactions on Intelligent Transportation Systems* 13(1): 385–394.

Wang, Y.; and Papageorgiou, M. 2005. Real-time freeway traffic state estimation based on extended Kalman filter: a general approach. *Transportation Research Part B: Methodological* 39(2): 141–167.

Wu, C.-H.; Wei, C.-C.; and Su, D.-C. 2004. Travel-time prediction with support vector regression. In *IEEE International Conference on Intelligent Transportation Systems*, volume 5, 276–281.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *International Joint Conference on Artificial Intelligence(IJCAI)*.

Yang, S.; and Qian, S. 2018. Understanding and Predicting Roadway Travel Time with Spatio-temporal Features of Network Traffic Flow, Weather Conditions and Incidents. *Transportation Research Board(TRB)* .

Yu, B.; Yin, H.; and Zhu, Z. 2018. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. In *International Joint Conferences on Artificial Intelligence(IJCAI)*.

Yu, E.; and Chen, C. 1993. Traffic prediction using neural networks. In *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM'93., IEEE*, 991–995. IEEE.

Zhang, J.; Zheng, Y.; and Qi, D. 2017. Deep Spatio-Temporal Residual Networks for City Wide Crowd Flows Prediction. In *The thirty-first Association for the Advance of Artificial Intelligence Conference(AAAI)*.

Zhou, C.; and Nelson, P. 2002. PREDICTING TRAFFIC CONGESTION USING RECURRENT NUERAL NETWORK. In *World Congress on Intelligent Transport System*.