Transportation Research Part C 121 (2020) 102877

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Congestion recognition for hybrid urban road systems via digraph convolutional network

Xiao Han, Guojiang Shen, Xi Yang, Xiangjie Kong *

College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China

A R T I C L E I N F O

Keywords:
Congestion recognition
Traffic state classification
Graph convolutional neural network
Intelligent transportation systems

A B S T R A C T

Congestion recognition is the prerequisite for traffic control and management, vehicle routing, and many other applications in intelligent transportation systems. Different types of roads with traffic facilities provide multi-source heterogeneous field traffic data, which contain the fundamental information and distinct features for congestion recognition. To exploit these traffic big data, in this paper, we propose a machine learning-based framework to tackle the congestion recognition problem. It can be divided in two parts, a digraph-based representation for hybrid urban traffic network and a Dirgraph Convolutional Neural Network (DGCN)-based learning model. At first, the representation incorporates the fundamental traffic variables with the correlation of different traffic flows, and partially decouples the global network topology from local traffic information. And then, to proceed with digraph-based samples, a new type of graph feature extraction method is introduced and the graph Fourier transform is defined accordingly. This distinguishes the proposed model from the conventional graph convolutional networks. Comprehensive experiments are conducted based on real traffic data. The results demonstrate the advantages of the proposed framework over the existing congestion recognition methods.

1. Introduction

It is crucial to recognize congestion in real-time for both the traffic management system and the drivers to ease the adverse effect of congestion, so that appropriate signal control strategy and vehicle driving strategy can be quickly performed to relieve the traffic congestion and restore the road traffic capacity. Traffic congestion is one of the trickiest problems in transportation system. The research on traffic congestion problem around great attention since the late 1950s, and is still an active topic in the fields of modern Intelligent Transportation System (ITS) and Internet of Vehicles (IoT) (Shen et al., 2018; Kong et al., 2020). Many researchers utilize these data in different ways to estimate traffic state. For instance, based on the inseparable correlation between traffic congestion and traffic state of local road network, C. Yuan et al. adopt variable-order Markov and probability suffix tree to extract the correlation rules and improve the overall traffic state prediction performance (Yuan et al., 2019). These studies mainly aimed to the classical theories based on the fundamental diagram of traffic flow have two phases: free flow and congested traffic which formed in the last century. While the static traffic states have a certain effect, in fact, it seem more important to capture the dynamic changes of the real road network traffic conditions nowadays. Inspired by the theory of three-phase traffic flow (Kerner, 1999), we would like to introduce the other two dynamic traffic states for two basic phases to describe the road situation more accurate: congestion formation state and congestion dissipation state.

* Corresponding author.
E-mail address: xjkong@ieee.org (X. Kong).

https://doi.org/10.1016/j.trc.2020.102877
Received 18 May 2020; Received in revised form 6 October 2020; Accepted 2 November 2020
Available online 27 November 2020
0968-090X/

At the same time, with the emerging techniques such as V2V, V2X and traffic big data, large volume of real-time traffic data are provided through the on-vehicle devices and other traffic facilities, which greatly facilitate the recognition or the prediction of traffic status, and the implementation of efficient routing algorithm or traffic management (Kong et al., 2019; Bauza and Gozalvez, 2013; Néstor et al., 2016; Junghans and Leich, 2016; Kachroo et al., 2017; Xiong et al., 2019). However, the current data analysis and processing methods with respect to these traffic big data still do not suffice the requirements of the traffic control and management, and the gap between the large traffic demand and the limited supply capability has been widen. Notice that there exists multiple types of road in the urban traffic network, such as the arterial road, auxiliary road, urban express-way, elevated road, etc., and different types of road also exhibit distinct traffic dynamical behaviours. Due to the inappropriate urban infrastructure planning or the imbalanced urban functional area layout, these roads are in fact interlaced in the urban traffic network, which result in the hybrid nature of the network, and greatly aggravate the complexity of traffic flow analysis. Most of the existing works only focus on single road type with certain regular structure for traffic states analysis and estimation, e.g., the grid structure for ground roads. For example, X. Chen et al. demonstrate that the proposed method can better capture the dynamic and variable information of expressway (single road type) in the process of congestion formation, spread and dissipation (Chen et al., 2019). In this scenario, the spatial geographic data can be simply reduced to a structured form such as a 1-dimensional array or a 2-dimensional matrix, and the inner spatial dependencies of hybrid network are neglected. However, the roads in urban traffic network are more complex. Although the traffic behaviors of different types of road are distinct, they still have strong correlations due to the traffic flow distribution through the adjacent sections among them. Thus the analysis of the characteristics and correlation of different traffic flow within the hybrid urban traffic network is still a crucial issue for congestion recognition problem.

To overcome the difficulties of the aforementioned problems and recognice dynamic traffic congestion states mentioned before, in this paper, we propose a framework to tackle the congestion recognition problem for hybrid urban traffic networks. In this framework, we first provide a novel digraph-based representation of hybrid urban traffic network. The nodes in the digraph contain the fundamental multi-dimensional traffic variables regarding certain traffic flows, and the direct edges represent the correlation between adjacent traffic flows. This kind of representation enables us to generalize multi-sources heterogeneous traffic data w.r.t. different types of roads into a unified digraph-based form, and partially decouple the global network topology from local traffic information. Then we establish a spanning tree generation technique, under which local traffic samples can be acquired at different locations and different time from the original field data. According to the digraph-based representation of the samples, both graph feature and node information include essential traffic characteristics for congestion recognition. Finally, we design a novel machine learning model to achieve real-time congestion recognition. The model is based on the Digraph Convolutional Neural Network (DGCN). In particular, to proceed with diagraph-based samples, a new type of graph feature extraction method is introduced, and the graph Fourier transform is further defined. Consequently, the convolution operation is performed with the graph feature and the node information with trainable model parameters to achieve the goal.

To the best of our knowledge, our work is the first attempt to introduce Graph Fourier Transform towards the digraph convolution method to tackle dynamic traffic congestion problems under hybrid urban traffic networks. Meanwhile, the extension of the graph convolutional neural network from undirected graph-based data to directed graph-based data is of great importance from the machine learning perspective.

The rest of the paper is organized as follows. In Section 2, the background of graph convolutional networks (GCNs) is introduced, and relevant works of both spatial-based and spectral-based GCNs are summarized. In Section 3, we formulate congestion recognition problem for urban traffic network, and provide the general framework for tackling this problem. In Section 4, the digraph-based representation of the traffic network is introduced. It facilitates us to describe the distribution pattern of the traffic flows based on the network topology, and quantify the essential traffic variables for congestion recognition. In Section 5, the design and implementation of the DGCN-based method is elaborated, the essential steps of sample acquisition, feature extraction and architecture of the DGCN learning model are provided. In Section 6, three types of experiments are conducted based on field traffic data, the effectiveness and efficiency of the proposed DGCN-based method are verified, and the comparative results show the advantages of the proposed method over the classic neural network-based and other GCN-based methods. The paper is concluded in Section 7.

Terminologies and Notations: Throughout the paper, the following terminologies and notations will be used.

A weighted directed graph, or digraph, is denoted by $\mathscr{D} = \{\mathscr{V}, \mathscr{E}, \mathscr{A}\}$. where $\mathscr{V}$ is the node (vertex) set, $|\mathscr{V}| = n$, $\mathscr{E}$ is the edge set, $\mathscr{A} \triangleq [a_{ij}] \in \mathbb{R}^{n\times n}$ is the weighted adjacency matrix of $\mathscr{D}$. In $\mathscr{D}$, a directed edge from node v_i to node v_j is denoted by $e_{i,j} = \ < v_i, v_j > \in \mathscr{E}$, the edge weight $a_{ij} > 0$ if $< v_i, v_j > \in \mathscr{E}$, otherwise $a_{ij} = 0$. If there exists a directed path from v_i to v_j, v_j is said to be reachable from v_i. The directed path from node v_i to node v_{i+k} can be denoted by $(v_i, v_{i+1}, \cdots, v_{i+k})$, where $< v_{i+l}, v_{i+l+1} > \in \mathscr{E}, l = 0, \cdots, k-1$. In the directed spanning tree $\mathscr{T} = \{\mathscr{V}_t, \mathscr{E}_t\}$ of a digraph $\mathscr{D} = \{\mathscr{V}, \mathscr{E}\}$, there exists only one directed path from the root node to any other nodes, and $\mathscr{V}_t = \mathscr{V}, \mathscr{E}_t \subseteq \mathscr{E}$.

In the paper, "traffic flow", or "flow", refers to the directed flow of the road vehicles on one or multiple lanes within a specified road section. The traffic flow is denoted by f_i, the traffic variables related to f_i include: the flow rate q_i (vehicles/h or vehicles/min), the average speed $v_{speed-i}$ (km/h or m/s), the vehicle density ρ_i (vhicles/km), the occupancy rate o_i (%), the traffic volume Q_i (vehicles/h), etc.

2. Background on graph convolutional neural networks

The research on graph-structured data in the field of machine learning traces back to early 2000s (Gori et al., 2005; Scarselli et al., 2009). These early studies mainly fall into the category of recurrent graph neural network-based model, which results in a great computational cost and cannot be directly applied in practice (Henaff et al., 2015; Niepert et al., 2016; Yu et al., 2017). With the rise of deep convolutional neural networks (CNNs), certain efforts have been paid to apply the convolution operation to graph-structured data, which result in the so-called graph convolutional neural networks (GCNs). These methods can be roughly divided into two categories, the methods based on aggregated space node information and the methods based on spectral theory, or alternatively, the spatial-based methods and the spectral-based methods. Spatial-based methods inherit ideas from those early studies by aggregating information to define graph convolutions (Scarselli et al., 2009; Henaff et al., 2015; Niepert et al., 2016; Yu et al., 2017; Li et al., 2015; Dai et al., 2018). Motivated by the convolution operation of CNNs on images, they also define graph convolution operation by nodes' spatial relations. They mainly convolves the target node's feature with its neighbors' features to derive the updated feature for the target node. As the pioneer work of spatial-based GCNs, Neural Network for Graphs (NN4G) (Micheli, 2009) performs graph convolutions by summing up a target node's neighborhood features directly. The layer node state of NN4G takes the following form:

$$h_v^{(k)} = f\left(x_v W^{(k-1)} + \sum_{i=1}^{k-1} \sum_{u \in N(v)} h_u^{(k-1)} \Phi^{(k-1)}\right), \tag{1}$$

where $f(\cdot)$ is an activation function and $h_v^{(0)} = 0$. $\Phi^{(k-1)}$ is a matrix of learnable parameters.

For the target node, since the number of its neighbors may vary from none to tens of thousands, it is impossible to get the full information of its neighbors, and the full-batch training algorithm for GCNs suffers from the memory overflow problem. To save memory, GraphSage (Hamilton et al., 2017) proposes a batch-training algorithm. It takes each node as a tree root node and expands the root node's neighborhood by K steps with a fixed sample size in a undirected graph.

However, the biggest problem with spatial-based methods is that they need to update the hidden layer state of all nodes in each layer, which will take up a lot of memory space. This problem may become fatal in the big data scenario such as dealing with massive traffic flow data.

On the other hand, spectral-based approaches have a solid mathematical foundation in the field of graph signal processing (Shuman et al., 2013; Sandryhaila and Moura, 2013; Chen et al., 2015). In most cases, the concerned graphs are assumed to be undirected, which can be represented by the corresponding normalized graph Laplacian matrices L. For undirected graph, it is evident that $L \in \mathbb{R}^{n\times n}$ is symmetric and positive semi-definite. By eigenvalue decomposition, the normalized graph Laplacian matrix can be factored as:

$$L = U \Lambda U^{\top}, \tag{2}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ is a diagonal matrix, λ_i is the eigenvalues of L, $U = [u_1, \cdots, u_n] \in \mathbb{R}^{n\times n}$, $u_i \in \mathbb{R}^n$ is the eigenvectors corresponds to eigenvalue λ_i, and $U^{\top}U = I_n$, i.e., the eigenvectors $u_i, i = 1, \cdots, n$ form an orthonormal space.

In graph signal processing, assume a graph signal $x \in \mathbb{R}^n$ is a feature vector where x_i is the value of the ith node. The graph Fourier transform is defined as $\widehat{x} = U^{\top}x$, and the inverse graph Fourier transform is defined as $x = U\widehat{x}$. With regard to undirected graph G, the spectral graph convolution of graph signal x with a filter $g \in \mathbb{R}^n$ is defined as:

$$x \bigstar_G g = U(U^{\top}x \odot U^{\top}g) = U(U^{\top}g \odot U^{\top}x), \tag{3}$$

where $\odot$ denotes the Hadamard product. Further, define the filter $g_\theta = \mathrm{diag}(U^{\top}g)$, then the spectral graph convolution can be simplified as $x \bigstar_G g_\theta \triangleq x \bigstar_G g = U g_\theta U^{\top}$.

The differences of existing types of spectral-based convolutional graph neural networks reflect on the choice and the approximation of the filter g_θ. For instance, in spectral convolutional neural network (Spectral CNN) (Bruna et al., 2014), the filter is considered as a set of learnable parameters as $g_\theta = \Phi_{i,j}^{(k)}$.

The research and applications of GCNs along with other kinds of graph neural networks have shown great success in numerous fields. However, for our concerned congestion recognition problem, there are still several issues need to be further explored. In spatial-based methods, there is no mathematical theory to prove that these methods are theoretically feasible. In spectral-based methods, the convolution operation is normally defined for undirect graph, so most of the existing methods are not directly applicable for the directed graph-based data such as the traffic variables related to directed traffic flows.

3. The framework for congestion recognition

The occurrence and the propagation of traffic congestion within the traffic network is observed by real measured traffic data. Fundamental traffic variables, such as vehicle speed v_{speed}, flow rate q, vehicle density ρ, occupancy rate o, are typically used as indicators to monitor the real-time traffic status and reveal the congestion. Meanwhile, the statistical indexes such as average traveling time, average time delay, and number of stops may assist to identify congestion and evaluate the traffic control performances. The collection and utilization of these traffic variables are essential for congestion recognition.

In large-scale hybrid traffic network, traffic flows are normally associated with complex dynamics that can be further categorized into different spatiotemporal patterns. The analysis of the real measured traffic data is crucial for understanding these patterns and further discovering certain traffic features to achieve pattern recognition. It is worth noting that within the traffic network, different types of roads associated with different traffic demands, different signal control levels, and different detection devices provide us with different contents and formats of traffic data. These multi-sources heterogeneous field data may create great challenge in data analysis to achieve satisfactory recognition results. Besides, with the expansion of the traffic network, the amount of field data dramatically

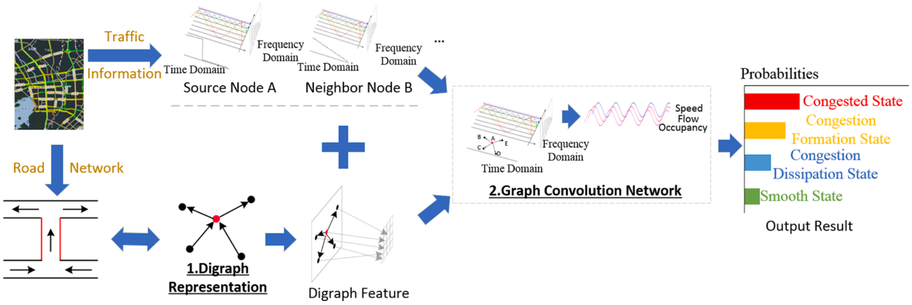


Fig. 1. The framework of congestion recognition.

increase and may exceed the computational capacity of the intelligent transportation system.

To tackle these challenging problems, in this paper, we propose a framework for real-time traffic congestion recognition in hybrid traffic network. The main ingredients of the framework include: 1. digraph-based representation of the traffic network, and 2. sample acquisition, feature extraction and congestion recognition through graph convolutional network-based model, see the right side of Fig. 1 for illustration.

In this framework, first, we establish a novel digraph-based representation of the hybrid traffic network. By this representation, the directed traffic flows within specified road sections are considered as the nodes, and the link from the upstream node to the downstream one are considered to be directed edges. More importantly, the correlation degree is defined for the adjacent nodes and served as the weight of the edge. It also can be extended to any two reachable nodes in the same directed path. Meanwhile, the multiple traffic variables of the concerned traffic flow are collected from the real field data and stored as the node information. This kind of representation facilitates us to maintain the fundamental traffic information, and partially decouples the global network topology from the local traffic characteristics in certain sense.

In the second step, a digraph convolutional network-based learning model is designed to achieve congestion recognition. Before the training process, we acquire the traffic samples from the historical database by using the digraph-based representation. Each sample is concerned with a target traffic flow, and by considering it as the root node, certain local directed spanning tree can be generated from the root node regarding the network topology and the distribution pattern of the target traffic flow. Further, utilizing the correlation degrees among all nodes in the spanning tree, the traffic information matrix can be defined to characterize the general effect of traffic distribution. Also, for the convenience of urban-scale applications, certain node ordering requirement and pruning techniques are utilized to retain the basic information and restrict the size of the samples. In turn, the acquired samples contain diverse spatiotemporal information, i.e., traffic variables collected in different locations at different time, which are vital for the training of the learning model. In the training procedure, the feature extraction is conducted w.r.t. the relevant traffic information matrices of different samples. A new kind of Fourier transform is introduced which enables us to implement the convolution operation on the digraph w.r.t. the node information. The convolutional results are further implemented through classifier, and assist to update the weight coefficients, i.e., parameters of the learning model. It is worth noting that this learning model is different from the conventional GCN-based ones since the graph convolution is proposed for digraph rather than undirected graph. see Fig. 1 for illustration.

4. The digraph-based representation of the hybrid traffic network

In this section, several fundamental definitions and characterizations utilized in the framework are proposed. To be specific, the categorization of traffic state provides the labels of the samples. The digraph-based representation of the traffic network, along with the definition of correlation degree, describes the distribution pattern of the traffic flows, and quantifies the essential traffic variables that collected from the field for the purpose of congestion recognition.

4.1. Categorization of traffic state

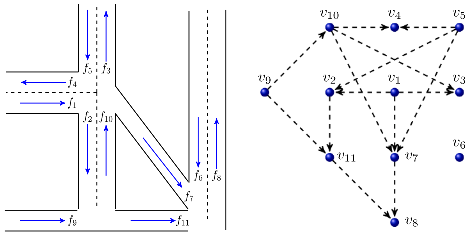
It is well conceived that, in general, the spatiotemporal distribution of traffic flow results in different traffic patterns, and congestion is one of them. Since vehicles need to know the occurrence or the forming stage of the congestion on the road ahead, in this paper, we categorize the traffic state into 4 different classes by using the empirical traffic data, namely, the congested state, congestion formation state, congestion dissipation state, and smooth state. Notice that the congested state and the smooth state are well perceived by most travelers and can be considered as the *inherent traffic patterns*. However, the subjective feelings of different people regarding the formation and dissipation of the congestion are not the same. Therefore, they are considered to be the *transient traffic patterns*, which may help to better describe the variations of the traffic.

It is worth mentioning that due to the different road types and signal control levels, these traffic states are to be defined w.r.t. specified traffic flow within the road section, which is either signal-controlled or signal-free. The “percentage of long-time-waiting vehicles” (p.LTW, %), i.e., the proportion of vehicles in the upstream section that wait for more than 2 cycle time of the intersection, and occupancy rate (o, %) are utilized as the indexes w.r.t. the signal-controlled and signal-free scenarios, respectively, see Table 1.

Table 1
Categorization of traffic states.

	Signal-controlled (p.LTW, %)	Signal-free (o, %)
Congested state	$\geqslant\alpha\%$	$\geqslant\gamma\%$
Congestion formation state	$\beta\% \nearrow \alpha\%$	$\kappa\% \nearrow \gamma\%$
Congestion dissipation state	$\alpha\% \searrow \beta\%$	$\gamma\% \searrow \kappa\%$
Smooth state	none of the above	none of the above

☆ The symbols $\nearrow$ and $\searrow$ mean "continuously increase or decrease".
☆ $\alpha, \beta, \gamma, \kappa$ is related to the citywide traffic situation of different cities. We define $\alpha = 50\%, \beta = 10\%, \gamma = 60\%, \kappa = 30\%$ for experiments after consulting the local traffic managers.



(a) (b)

Fig. 2. A sample of digraph representation.

As Table 1 shows, we cannot judge congestion formation state and congestion dissipation state in real time. Because both states require continuously increase or decrease of p.LTW. However, we can mark these traffic states on historical data. Therefore, our aim is to find a way that can identify all traffic states in real time.

4.2. *Digraph-based representation of traffic network*

For traffic network, it is conventional to utilize an undirected graph to represent the topology and certain traffic information, where the nodes represent the intersections, the edges represent the road sections, and the related traffic information such as the distance, average speed, traffic column, etc. can be considered as the weights of the corresponding edges.

Normally, most roads in urban traffic network permit two-way traffic, and the characteristics of the traffic flow in the same road section but with opposite directions are quite the same, so the aforementioned undirect graph representation is adequate with the conventional methods. However, it is worth noting that there still exists one-way street or parallel roads between two intersections, and the phenomenon of tidal traffic are frequently occurred in urban traffic network. In these scenarios, the undirected graph representation of the traffic network will not suffice. More importantly, as stated in 4.1, we aim to recognize the congestion that occurs on the road section rather than at the intersection according to collected data regarding the traffic flow. It is evident that the traffic flow and its distribution is directed, so it is possible to alternate the conventional undirected graph-based traffic network representation by considering the traffic flow within a road section as the node and using the directed edge to indicate the distribution of the traffic flows.

Example 1. Consider a sample road network illustrated in Fig. 2(a). The traffic flows within the network are labeled from f_1 to f_{11}. It is assumed that at the intersections, the vehicles are permitted to go straight, turn left or right, but not permitted to turn around. In Fig. 2(b), the digraph-based representation of the sample road network is given. In this digraph, node v_i corresponds to traffic flow f_i, and the directed edge represents the permitted direction of traffic flow distribution so that the upstream and downstream relation of different traffic flows are explicitly exhibited. Also, the direct edges implicitly indicate the intersections between relevant road sections which may be signal-controlled or signal-free. □

The aforementioned digraph-based representation is initialized w.r.t. the distribution pattern of the traffic flow, and it partially decouples the local traffic information from the global network topology, so that we can focus ourselves to the collected field data such as flow rate, average speed and occupancy rate, and consider them as the high-dimensional node information. The digraph-based representation also unifies different types of roads with different signal control levels in the traffic network, such as the signal-controlled or signal-free road, one-way street, elevated road and the ramp metering expressway. The inconvenience caused by using the undirected graph such as the parallel roads or the phenomena of tidal traffic is released.

In the next subsections, we complete the digraph-based representation by quantifying the correlation degrees of the successive traffic flows and defining them as the weights of the directed edges. In turn, both node information and edge information of the digraph contain essential traffic variables and indexes.

4.3. Definition of correlation degree

In this subsection, we define the correlation degree of the successive traffic flows. It is worth noting that in the digraph-based representation, the directions of the edges exhibit the distribution pattern of traffic flows, and the correlation degrees will further characterize the influence strength between the traffic flows. For simplicity, consider the upstream traffic flow f_i and downstream one f_j. We denote $d_{i,j}$ as the correlation degree of f_j w.r.t. f_i, and use it as the weight of the directed edge $e_{i,j}$, i.e., $a_{i,j} = d_{i,j}$.

Besides, it is well conceived that the signal control level of the intersection between adjacent road sections definitely effects the correlation degree of the successive traffic flows. Correspondingly, the correlation degree is defined for the signal-control and signal-free intersections as $d_{i,j}^{\text{sig}}$ and $d_{i,j}^{\text{n-sig}}$, respectively.

4.3.1. Signal-free case

When there is no signal-control intervention between traffic flows f_i and f_j, the real-time correlation degree is defined as follows,

$$d_{i,j}^{\text{n-sig}} = (1 - \frac{|\Delta q_i - \Delta q_{i\to j}|}{C_{i,j}}) \cdot \delta \tag{4}$$

where q_i and $q_{i\to j}$ are the flow rates, Δq_i and $\Delta q_{i\to j}$ denotes the variations of the flow rates within the unit time forward from the current moment, and $C_{i,j}$ is the maximum admissible increment of traffic volume that allows to transfer from f_i to f_j within the unit time. δ stands for the environmental impact factor, $0 < \delta < 1$.

4.3.2. Signal-controlled case

When f_i and f_j are connected by a signal-controlled intersection, the real-time correlation degree is defined for the successive traffic flows that have the r.o.w. during the whole signal cycle. For simplicity, assume the time of the corresponding green phase is $[0, T_g]$, and the average time to empty the waiting vehicle queue is T, then

$$d_{i,j}^{\text{sig}} = \begin{cases} \widetilde{d}_{i,j} & \text{if} \quad t \in [0, T_g], \quad t \geqslant T \\ \frac{t}{T}\widetilde{d}_{i,j} & \text{if} \quad t \in [0, T_g], \quad t < T \\ 0 & \text{if} \quad t \notin [0, T_g] \end{cases} \tag{5}$$

where t is the current time, $\widetilde{d}_{i,j}$ denotes the correlation degree calculated by (4), i.e., by considering f_i and f_j as the free flows without signal-control. This indicates the fact that the effect of signal control period on the correlation degree must be taken into consideration for the signal-controlled case.

The correlation degree can be further extended to more general case where the traffic flows are not adjacent. As a matter of fact, it is natural to recognize the upstream traffic flow is always distributed through multi-level of sections.

Consider traffic flow f_i and f_j with the corresponding nodes v_i and v_j in digraph $\mathscr{D}$. If v_j is reachable from v_i, then there exists at least one direct path from v_i to v_j. For convenience, denote all possible paths as $p^{(1)}, \cdots, p^{(n)}$, then along the path $p^{(k)}, k \in \{1, \cdots, n\}$, the correlation degree of f_i and f_j is defined as follows,

$$d_{i,j}^{(k)} = d_{i,i+1}d_{i+1,i+2}\cdots d_{j-1,j}, \tag{6}$$

where $p^{(k)} = (i, i+1, \cdots, j-1, j)$. Consequently, the correlation degree of f_i and f_j is defined as the maximal correlation degree considering all the possible paths, i.e.,

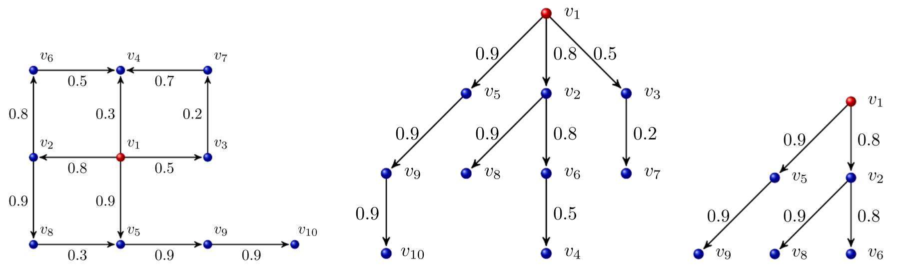
$$d_{i,j} = \max\{d_{i,j}^{(1)}, \cdots d_{i,j}^{(n)}\}. \tag{7}$$

where we use the max function instead of the sum function. Because for each vehicle traveling from the starting point to the end point, it is impossible for this vehicle to reach the target through multiple paths at the same time on the same trip.

It is quite evident that due to the digraph-based representation and the definition of correlation degree, the distribution pattern and the influence strength among traffic flows, either they are adjacent or not, are appropriately characterized.

5. The digraph convolutional network-based learning model

In this section, we elaborate the design and implementation of the Directed Graph Convolutional Neural Network (DGCN) learning model to tackle the congestion recognition problem. First, we collect traffic samples from the traffic network data. Each sample is acquired by generating a local spanning trees with a specified traffic flow as the root node. The digraph-based representation of the sample contains the relevant traffic variables of each node, and exhibits the distribution pattern and the influence of the concerned traffic flow. The current state w.r.t. this specified traffic flow is also labeled with the sample. Second, we extract graph features from each sample. In particular, to proceed with these digraph-structured data, a new kind of diagonalization technique is proposed w.r.t. the traffic information matrix of each sample, and a new type of graph Fourier transform is defined accordingly. This distinguishes the



(a) Detailed Digraph (b) Spanning Tree Generation (c) Pruning Result ($K = 3$, $N = 6$)

Fig. 3. A traffic control area.

proposed method from the conventional methods which are only applicable for the undirected graph. And finally, the structure and the implementation of the DGCN learning model is provided. During the training procedure, the labeled samples are used as the input, and the output of the model is processed through a multi-class classification problem, under which the model parameters are updated. After training, the learning model is capable to tackle the traffic state classification problem, and thus the congestion recognition is achieved.

5.1. Samples acquisition by local spanning tree generation

As previously mentioned, we focus ourselves on the recognition of congestions that occurred on the specific road sections. To this end, we realize that the traffic flow within local sections contains almost all the useful information for congestion recognition, e.g., the traffic variables and the effect of distribution. So it is possible for us to acquire samples w.r.t. certain traffic flows in a small traffic area.

To be specific, for the traffic network represented by digraph $\mathscr{D}$, by selecting a target node v_r as the root, it is possible to generate a spanning tree $\mathscr{T}$ that includes all reachable nodes from v_r. In particular, since there may exist more than one directed path from v_r to other nodes, the formation of the spanning tree is not unique. To better exhibit the effects of traffic flow distribution w.r.t. the root node, for node v_j in $\mathscr{T}, j \neq r$, its parent node is defined as the predecessor node along the directed path under which v_j achieves its correlation degree w.r.t. v_r, see Eq. (7) and the following example for illustration.

More importantly, to confine the size of the sample, a sub-spanning tree $\mathscr{T}_s$ can be further derived under two predefined parameters N and K, where N is the number of nodes and K is the number of layers in $\mathscr{T}_s$. For convenience, the child nodes with the same parent node are arranged from left to right according to the correlation degree w.r.t. the parent node. See the following example.

Example 2. Consider a small traffic control area represented by the digraph in Fig. 3(a). To acquire a sample, we first select a target node, namely, v_1, and consider it as the root node. By traversing all nodes in the digraph, we realize all nodes are reachable from v_1, and apparently, there may be more than one direct path from v_1 to other nodes, e.g., for node v_4, there exist three directed paths (v_1, v_4), (v_1, v_2, v_6, v_4) and (v_1, v_3, v_7, v_4). In this case, as defined in Eq. (7), the correlation degree of v_4 w.r.t. v_1 is 0.32 according to the path (v_1, v_2, v_6, v_4), so that v_4 is considered to be the child node of v_6 rather than v_1 or v_7. In turn, a spanning tree with node v_1 as the root can be generated, c.f. Fig. 3(b). Finally, we empirically determine the parameters $N = 6$ and $K = 3$, and it yields a sub-spanning tree as in Fig. 3(c).

Notice that the nodes in the sub-spanning tree are with the highest correlation degree under the constraints of N and K. However, some nodes with higher correlation degree are discarded, e.g., v_{10} is not included in Fig. 3(c). The trade-off between the sample size and sample efficiency will be further discussed based on the experimental results in Section 6. □

5.2. Feature extraction of traffic information matrix

Based on diagraph-based representation, each sample can be denoted by $\mathscr{T}_s = \{\mathscr{V}, \mathscr{E}, \mathscr{A}\}$, where $|\mathscr{V}| = N$. For simplicity, rename all nodes in sub-spanning tree $\mathscr{T}_s$ from $\bar{v}_1$ (the root node) to $\bar{v}_N$ according to their appearance order from layer 1 to layer K. It can be realized that in $\mathscr{T}_s$, the correlation degree of any pair of nodes satisfies $d_{i,j} \in (0, 1]$ if there is a directed path from $\bar{v}_i$ to $\bar{v}_j$, otherwise $d_{i,j} = 0$. Then define a new kind of *correlation degree matrix* $M_c = [m_{i,j}] \in \mathbb{R}^{N \times N}$, where $m_{i,j} = d_{i,j}$. Notice that M_c is in the upper triangular form where all diagonal elements are 0. Furthermore, define the *degree matrix* $D_c = \mathrm{diag}[d_i] \in \mathbb{R}^{N \times N}$, where $d_i = \sum_{j=1}^{N} d_{i,j}$, and the *traffic information matrix* $T_c = D_c - M_c$. It is worth noting that the traffic information matrix is in the form of the Laplacian matrix of $\mathscr{D}$.

For the defined traffic information matrix T_c, along with the correlation degree matrix M_c and degree matrix D_c, it is evident T_c is also an upper-triangular matrix with diagonal elements being d_i. It is easy to show that eigenvalues of T_c satisfy $\lambda_i = d_i, i = 1, \cdots, N$, since the characteristic polynomial of T_c is $p(\lambda) = (\lambda - d_1)(\lambda - d_1)\cdots(\lambda - d_N)$.

However, since T_c is not a symmetric matrix, it is not possible to use eigenvalue decomposition-based method to extract feature vectors as introduced in Section 2 w.r.t. traffic information matrix T_c.

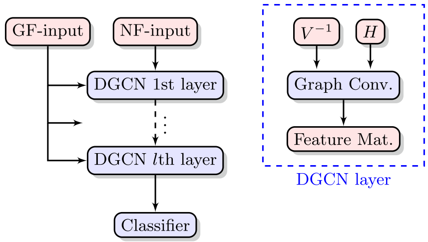


Fig. 4. The DGCN learning model.

Inspired by the recent works on graph convolution w.r.t. digraph-based data and graph Fourier transform w.r.t. digraph (Singh et al., 2016; Sardellitti et al., 2017; Shafipour et al., 2019), we implement the decomposition w.r.t. traffic information matrix T_c in the following form,

$$T_c = VD_cV^{-1} \tag{8}$$

where the invertible matrix $V \in \mathbb{R}^{N\times N}$ is in upper triangular form, and it can be numerically solved under the following formula,

$$\begin{aligned} &v_{i,j} = 1, \quad 1\leqslant j\leqslant N,\ \ i = 1 \\ &v_{i,j} = 0, \quad j < i,\ \ 2\leqslant i\leqslant N \\ &\sum_{j=i+1}^{k} t_{i,j}v_{j,k} = (t_{k,k} - t_{i,i})v_{i,k}, \quad 1\leqslant i < k \end{aligned} \tag{9}$$

where $v_{i,j}$ and $t_{i,j}$ are elements of V and T_c, respectively.

It is known that the graph feature relies on the vectors of the invertible matrix V, so V is referred to as the graph feature matrix. The existence of V guarantees us to define a new kind of graph Fourier transform pair $\widehat{x} = V^{-1}x$ and $x = V\widehat{x}$, and the graph convolution operation holds.

Example 3. Consider the sample exhibited in Fig. 3(c). We rename the nodes $v_1, v_5, v_2, v_9, v_8, v_6$ as $\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_6$. Then the correlation degree matrix, degree matrix and traffic information matrix are given as follows,

$$\begin{aligned} M_c &= \begin{bmatrix} 0 & 0.9 & 0.8 & 0.81 & 0.72 & 0.64 \\ 0 & 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9 & 0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ D_c &= \mathrm{diag}([3.87, 0.9, 1.7, 0, 0, 0]), \\ T_c &= \begin{bmatrix} 3.87 & -0.9 & -0.8 & -0.81 & -0.72 & -0.64 \\ 0 & 0.9 & 0 & -0.9 & 0 & 0 \\ 0 & 0 & 1.7 & 0 & -0.9 & -0.8 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned}$$

According to Eq. (9), the invertible matrix V can be numerically solved as follows,

$$V = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 3.3 & 0 & 2.2632 & 0 & 0 \\ 0 & 0 & 2.7125 & 0 & 1.7917 & 1.7917 \\ 0 & 0 & 0 & 2.2632 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3.3843 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.8073 \end{bmatrix}.$$

It can be verified that V satisfies Eq. (8). □

5.3. DGCN model

In this subsection, we elaborate the traffic digraph convolutional neural network (DGCN)-based learning model. The architecture of DGCN learning model and DGCN layer is exhibited in Fig. 4.

As shown in Fig. 4, the learning model is built by multiple DGCN layers. The input of each layer consists of the graph feature matrix, which is defined in the last section, and the matrix $H^{(l-1)}$ regarding the node feature, which satisfies

$$H^{(l)} = \sigma\left(V^{-1}H^{(l-1)}W^{(l)}\right), \tag{10}$$

where l is the layer index, $l = 1,\cdots,k$, N is the number of nodes in the sub-spanning tree-based sample, $V \in \mathbb{R}^{N\times N}$, $H^{(0)} = X, X$ is the node feature matrix of the sample, $H^{(l-1)} \in \mathbb{R}^{N\times f_{l-1}}$, f_{l-1} is the dimension of the node feature as the number of the input channels, f_l is the dimension of the node feature as the number of the output channels, $W^l \in \mathbb{R}^{f_{l-1}\times f_l}$ contains the trainable parameters in the lth layer, and $\sigma(\cdot)$ represents the sigmoid function for a nonlinear model.

For classification problem, denote the output, i.e., the extracted features, of the lth layer by $F^{(l)} = H^{(l)} \in \mathbb{R}^{N\times f_l}$. According to softmax function and cross-entropy loss for multi-class classification problem, in the classifier layer,

$$\begin{aligned} P_j &= \frac{e^{a_j}}{\sum_{k=1}^{4} e^{a_k}}, \ \ a_k = \theta_k^\top f^{(l)}, \ \ j = 1,\cdots,4, \\ L &= -\sum_{j=1}^{4} S_j \log(P_j), \end{aligned} \tag{11}$$

where P_j is the probability that the output belongs to class S_j, S_j denotes the true class label, $\widehat{S}_j$ denotes the estimated class label, $f^{(l)} = \mathrm{vec}(F^{(l)}) \in \mathbb{R}^{\ell}, \ell = N\times l$, $\theta_k \in \mathbb{R}^{\ell}$ are model parameters to be trained, $k = 1,\cdots,4$. and L is the cross-entropy loss.

Algorithm 1. Training of DGCN Learning Model

Input: Graph Feature GF, Node Feature NF, Traffic State S

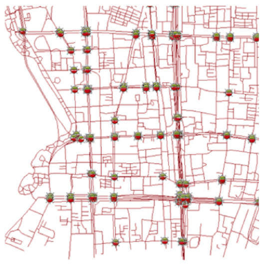
Output: Fixed Model Hyperparameters W

1 **for** *each pair* $(<GF_i, NF_i>, S_i)$ *of input,* **do**
2   **for** $l < k$ *(the maximal layer number),* **do**
3   Find the best hidden layer output $H^{(l)}$ w.r.t. $<GF_i, NF_i>$,
4         $l = l + 1$.
5   **end**
6   Calculate the overall model output $\hat{S}_i = H^{(k)}$, find the error $e_i = S_i - \hat{S}_i$,
7   **if** $|e_i| > e_{\min}$, **then**
8   Define the error function $E(<GF_i, NF_i>) = -\sum_{j=1}^{4} S_j \log(P_j)$.
9   Adjust the weights by $W_i^+ = W_i - \Delta$, $\Delta = \eta\frac{\partial E(<GF_i,NF_i>)}{\partial W_i}$.
10   **if** $|\Delta| \geq \delta_{\min}$, **then**
11   **return** step 1,
12   else
13   break
14   end
15   **end**
16 **end**

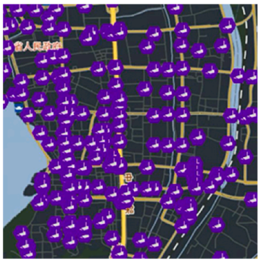
The training of DGCN learning model is given in Algorithm 1, and the basic procedures are summarized as follows. In the beginning, hyperparameters are randomly initialized. The graph feature matrix and node feature matrix are used as the input in order to proceed graph signals of different minibatch size. Then the error estimation is performed between the output of estimated class label and true label, meanwhile, the hyperparameters of the model are optimized for each epochs in the gradient descent manner.

The DGCN learning model is capable to autonomously discover and iteratively learn a filter with universal significance. In turn, the DGCN learning model can acquire the graph features related to congestion from different graph samples represented by different graph structures, i.e., the model can tackle the different distribution patterns and correlation dependence of the traffic flows. The model also provides us an alternative convolutional operation for digraph-based data, by with the local graph features combined with the node information jointly describe the traffic congestion information.

(a) Road Network & GPS Data



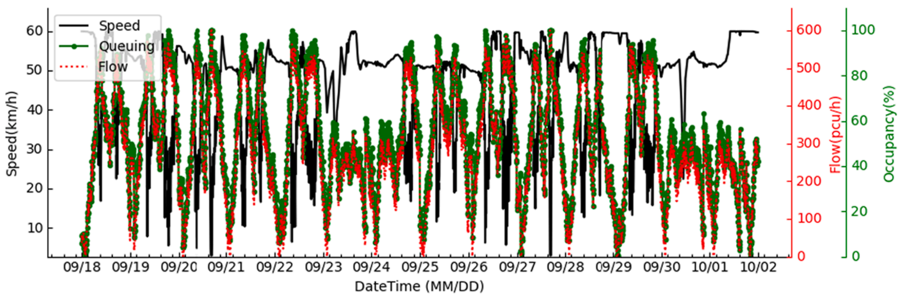
(b) Potision of Singal Lamps



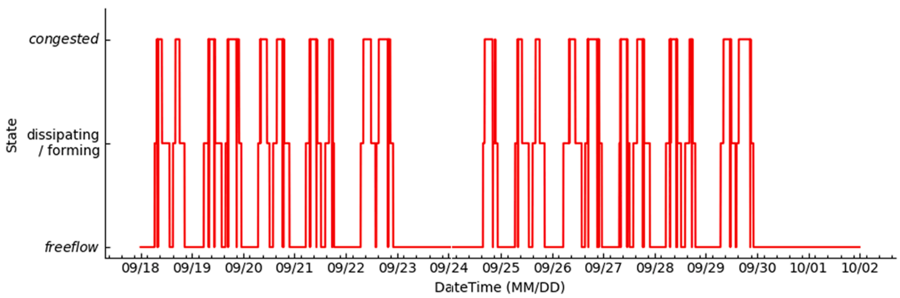
(c) Potision of Traffic Dectors

(d) Congestion Heatmap

Fig. 5. Original traffic data.



(a) Traffic Characteristic Parameters



(b) Traffic States

Fig. 6. Experimental data.

6. Experiments

6.1. Experiment Setting

Dataset: The traffic network under investigation covers the area of Longxiangqiao, Hangzhou, China. This area contains 1538 directed road sections which are the used to construct training data samples in different geographic locations. Based on the road centerline data, the network diagram is illustrated in Fig. 5(a).

The dataset contains the multi-source traffic data collected by about 9000 taxis in Hangzhou within half month (Sept. 18th – Oct. 2nd, 2017). The GPS data are uploaded by taxi every 40 s on average, and the average daily records from all taxis are more than 10 millions pieces. As an example, Fig. 5(b) shows all signal lamps in this road network, which can provide signal start time, signal period and other signal related parameters. Fig. 5(c) shows the position of cameras in the road network, which generate traffic data that is used to assist in the calculation of traffic parameters. Fig. 5(a) & (d) illustrates the collected GPS data and the real-time road congestion status during 8:00 AM to 8:01 AM on Sept. 18th, 2017.

Data Preprocessing: At first, Map matching method (Zhang and He, 2018) is used to map the low-sampling raw GPS data to the road segment and generate the driving trajectory of each taxi. Then carry out data cleaning: removing the trajectory data beyond the experimental road network. Then the road occupancy rate is estimated by the proportion of vehicles waiting at the intersection (in this data set, vehicles with a resting time of more than 120s are selected). Traffic detectors collect the flow data directly, and the roads without those static detectors pass the weighted statistics to estimate the road traffic flow of the vehicles passing by during the time interval. The average traveling speed of all vehicles passing through the road within the time period is calculated to obtain the average speed of road sections. The traffic flow data of one road segment is shown in Fig. 6(a). Then according to the *transient traffic patterns* defined in Section 4.1 of the article, we calculate the historical data label, as an example shown in Fig. 6(b). Finally, according to the sample acquisition method introduced in Section 5.1, we obtain about 6000,000 samples.

In the experiments, we divide these sample data into training dataset and test dataset at a ratio of 3: 1. For the training dataset, 25% data are used as training data, and 75% are used as validation data.

Comparison Methods: To verify the effectiveness of the proposed framework, and to evaluate the performance of the proposed learning model, three types of experiments are performed.

First, series of experiments are conducted by using the proposed DGCN learning model but with different sample parameters, i.e., K, N and LN. This type of experiments show the effects of sample parameters of the proposed method.

Second, our proposed method is compared with some classic machine learning-based learning methods. This type of experiments provides a comprehensive comparison of DGCN with the well-known machine learning methods such as AMSVM (Feng et al., 2018), HMM (Qi and Sherif, 2014), original CNN, multi-CNN (Li et al., 2020), DCRNN (Li et al., 2017), FCL-NET (Ke et al., 2017) and ST-ResNet (Zhang et al., 2018).

Third, the DGCN model is compared with some state-of-the-art graph convolutional network-based models and their variants (Kipf and Welling, 2017; Kejani et al., 2020; Yua et al., 2020; Yao et al., 2018; Wu et al., 2019). This type of experiments show the advantages of the extension of convolution operation on digraphs.

Performance Evaluation Indexes: Consider that the congestion recognition is in fact converted into traffic state classification, in this scenario, to evaluate the performances of different methods with their variants, the macro-averaging based evaluation indexes “Accuracy”, “Precision”, “Recall” and “F-Measure” (F1-score) are used for this multi-class classification problem. If the positive sample is P and the negative sample is N, then these indexes can be calculated as follows:

$$\begin{cases} Precision = \dfrac{TP}{TP + FP}, \\ Recall = \dfrac{TP}{TP + FN}, \\ Accuracy = \dfrac{TP + TN}{P + N}, \\ F - Measure = \dfrac{2}{(1/Precision) + (1/Recall)}, \end{cases} \tag{12}$$

where TP stands that the true value of the positive sample is the same as the predicted value, TN means that the true value of the positive sample is different from the predicted value, FP stands that the true value of the negative sample is the same as the predicted value, and FN means that the true value of the negative sample is different from the predicted value.

It can be seen from Eq. (12) that the four indicators have a mutually restrictive relationship, especially when the data samples are unbalanced. For example, most of the time in a road is in smooth state, and congestion generally only occurs during morning and evening rush hours. When the model is not trained and may fully predict that the samples are in smooth state, it still has high precision, but other indicators are low. Therefore, our goal is to balance various indicators and train a better parameter to ensure maximum model efficiency.

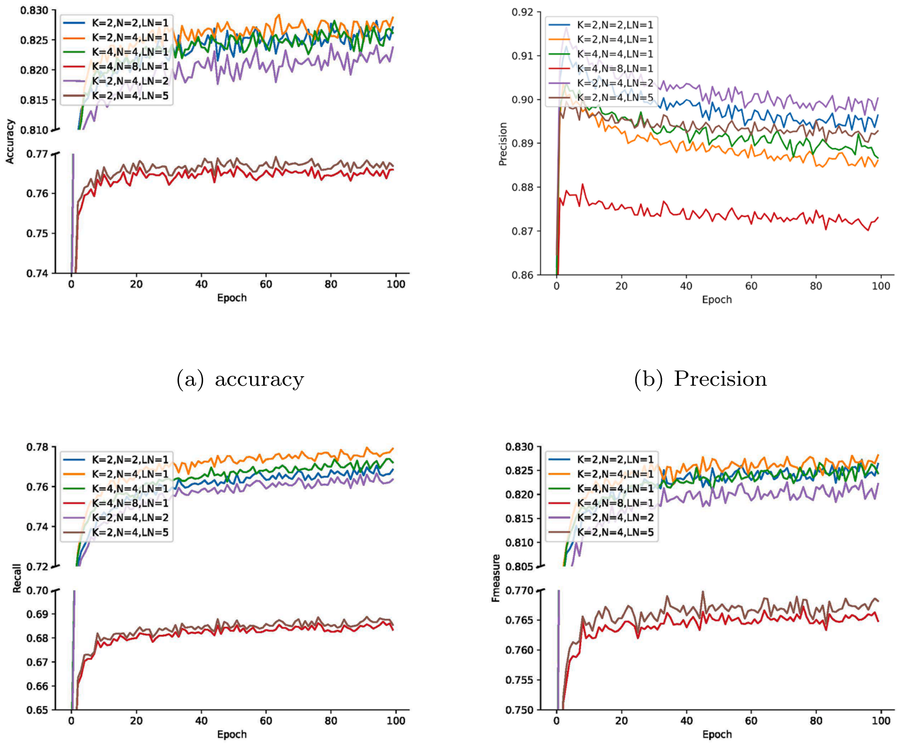
Fixed Parameters: The DGCN learning model is trained by using Adam optimizer with a learning rate 0.01 for 1024 epochs. The dropout technique is applied to all feature vectors with the rate 0.5. The channels for each layers in DGCN model is set to be 30.

6.2. Experimental results

We compare the proposed model DGCN with some state-of-the-art methods. These comparisons belong to three categories of tasks: different hyper-parameters of proposed methods, classic neural network-based methods and other graph-based semi-supervised methods, as Table 2 shows.

Table 2
The comparison of experimental results.

	Accuracy(%)	Precision(%)	Recall(%)	Fmeasure(%)
DGCN($K=2, N=2, LN=1$)	82.42	89.60	76.35	82.48
DGCN($K=2, N=4, LN=1$)	**82.52**	88.62	**77.92**	**82.68**
DGCN($K=4, N=4, LN=1$)	82.41	89.01	76.98	82.42
DGCN($K=4, N=8, LN=1$)	76.19	87.21	68.15	76.51
DGCN($K=2, N=4, LN=2$)	82.01	89.97	76.06	82.61
DGCN($K=2, N=4, LN=5$)	76.39	89.24	68.29	76.53
AMSVM (Feng et al., 2018)	69.65	81.05	72.16	72.03
HMM (Qi and Sherif, 2014)	74.13	77.85	66.95	73.56
multi-CNN (Li et al., 2020)	72.48	73.55	72.96	67.64
DCRNN (Li et al., 2017)	79.65	86.03	73.98	78.21
FCL-NET (Ke et al., 2017)	77.05	83.98	74.93	77.02
ST-ResNet (Zhang et al., 2018)	73.55	79.32	74.97	74.76
GCN (Kipf and Welling, 2017)	73.56	82.65	66.90	73.81
GCNMR (Kejani et al., 2020)	80.05	86.72	74.08	80.01
expanded-GCN (Yua et al., 2020)	81.98	88.13	76.80	82.00
DMVST (Yao et al., 2018)	79.21	85.09	74.01	79.01
Graph WaveNet (Wu et al., 2019)	80.99	87.78	75.05	80.95
DGCN(Dis-Mat)	81.65	**90.21**	74.10	81.41



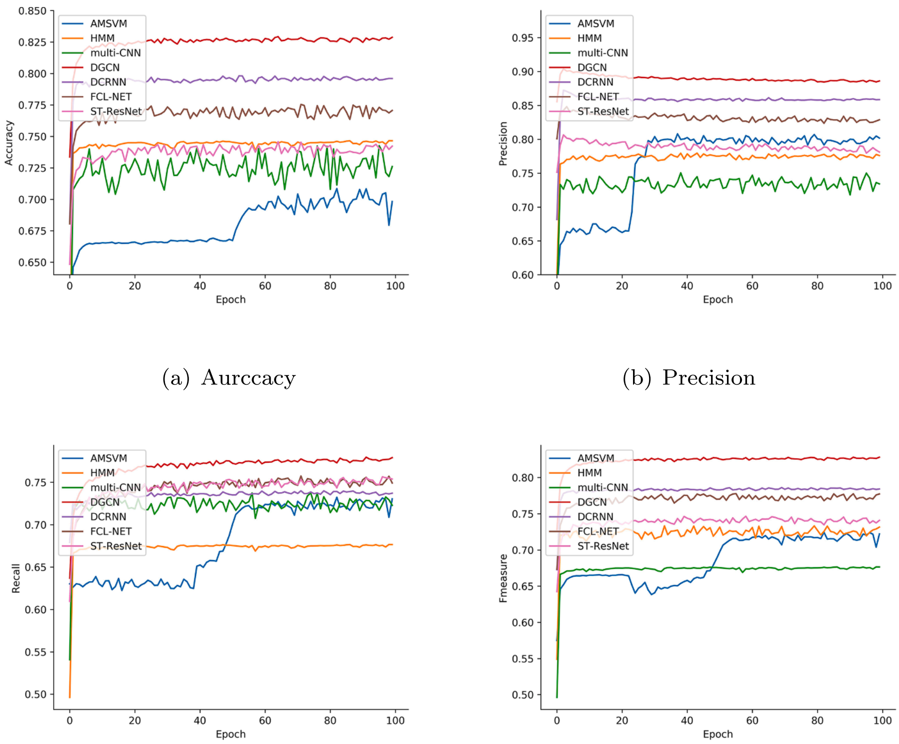
(a) accuracy (b) Precision

(c) Recall (d) F-measure

Fig. 7. Performance w.r.t. different hyper-parameters.

6.2.1. Effects of different hyper-parameters

As mentioned in Section 5.1, the hyper-parameters N and K need to be manually specified in sample collection procedure. Recall that N limits the total number of nodes in the sub-spanning tree which in turn determines the size of the sample, and K specifies the maximum depth of the sub-spanning tree that partially reflects the influence of the target traffic flow. The changing of N and K obviously alters the sub-spanning tree generated from the same root node, so that the resulting samples are different. Besides, different layer numbers LN, may also have a huge influence on the experimental results. With the proposed DGCN learning model, samples with different K, N and LN are used, and the experimental results are shown in Fig. 7(a)–(d).



(a) Aurccacy (b) Precision

(c) Recall (d) F-measure

Fig. 8. Different NN-based methods.

For the proposed DGCN-based method, it can be recognized that the general performance is not closely related to the pre-set parameters when they are small. For instance, with the same number of layers, the Accuracy and the F-measure of the first three combination of other hyper-parameters, i.e., $K = 2, N = 2$ or $K = 2, N = 4$ or $K = 4, N = 4$, are nearly the same, and the results of Precision and Recall are somehow contradictive as is well known in the classification problems. However, when the values of K and N are chosen to be large, e.g. $K = 4, N = 8$, there's a significant decrease in system performance.

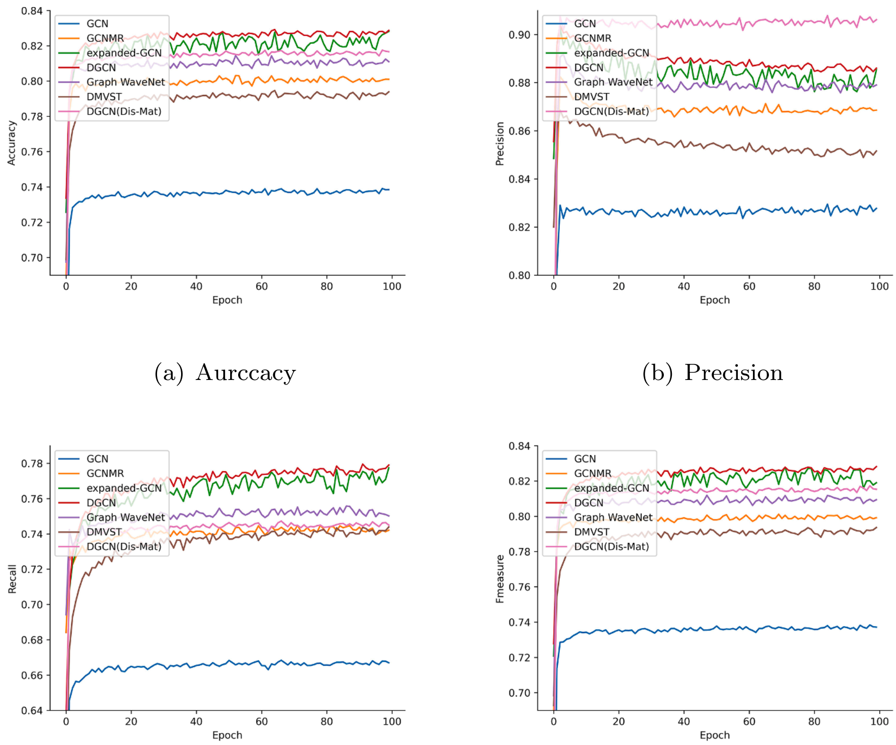
These facts mainly related to the characteristics of the original traffic data and the correlation degrees which are calculated based on these data. Indeed, it can be recognized that there are much more ground roads in the urban traffic network compared with elevated roads, urban expressway, etc., so the data collected from the ground roads dominate the dataset. Meanwhile, a big number of the ground roads are signal-controlled, so the correlation degree between the adjacent upstream and downstream traffic flows is not high. When the traffic flows are distant, i.e., when large K is chosen, the correlation degree of the downstream traffic flow w.r.t. the target one will dramatically decrease. Also, when large N is chosen, one may introduce quite a few number of "virtual nodes" with the correlation degree of 0 in the generation of sub-spanning tree. In these circumstances, the correlation degree matrix, and thus the traffic information matrix, can not appropriately reflects the inherent relations among traffic flows, and the sample quality is degenerated. Particularly, the introduction of "virtual node" significantly weakens the effectiveness of the proposed method, c.f. the case of $K = 4, N = 8$.

Moreover, it can be intuitively observed that the best performance achieves for the case $K = 2, N = 4$, see Fig. 7(a) and (d) for instance. We see that when $K = 2, N = 2$ or $K = 4, N = 4$, the resulting sub-spanning tree is merely a chaingraph, which contains less distribution pattern of traffic flows than the case $K = 2, N = 4$. From this perspective, it is necessary to balance the size and the ratio of N and K, so that the number of concerned downstream traffic flows and the distribution pattern w.r.t. the target traffic flow can be fully recovered in the sample, and the inherent relation of different traffic flows can be better characterized.

At the same time, when K and N are the same, it can be observed that the performance of the neural network does not increase with the deepening of the model depth, but has a greater decline. This may be because the spectral domain graph convolutional neural network can effectively filter a large number of features and simplify the data in the form of self-learning filters. However, as the number of network layers deepens, that is, superimposing different filters, some key features will also be affected. Abandon, eventually leading to a decrease in the accuracy of the experimental results.

6.2.2. Comparison with classic machine learning methods

We compare our method with three methods which are based on classic machine learning methods (include some other neural networks), namely, AMSVM, HMM, multi-CNN, DCRNN, FCL-NET and ST-ResNet. Since the mechanism and network structure of these



(a) Aurccacy (b) Precision

(c) Recall (d) F-measure

Fig. 9. Different NN-based methods.

methods are different, it is necessary to meet the uniformity requirement of the input data as much as possible.

In the experiments, all samples for the proposed DGCN are further proceeded for other machine learning models. For each sample, all nodes in the sub-spanning tree are sorted in the deceasing order according to their correlation degrees w.r.t. the root node. Consequently, a new *node information sequence* can be formulated, where each node contains the relevant traffic variables, such as speed (s), flow rate (f), occupancy rate (o), etc., and the traffic state defined for the sample is inherited as the label for the sequence. To clarify this node information sequence from the sub-spanning tree, we denote it by $(n_1, \cdots, n_N)$, where n_i contains the traffic information (s_i, f_i, o_i). The node information sequences along with its label will serve as the training data for the comparative NN-based methods.

For two traditional machine learning methods: AMSVM and HMM, it is convenient to use the node information sequences with labels as the training data to achieve the classification task in a supervised learning manner. In other NN-based methods, a 1-dimensional convolution kernel of size $1 \times a, a < N$, is used to perform convolution operation w.r.t. the node information sequence, and once again, the softmax layer is utilized in the network architecture to achieve classification.

For the proposed DGCN-based method and the aforementioned six kinds of classic neural network-based methods, the comparative results are shown in Fig. 8(a)–(d).

It is evident that DGCN-based method outperforms all other six classic neural network-based methods in almost all the performance indexes except the Precision, see the red line in all figures. The improvement of DGCN-based method is mainly attributed to its utilization of the graph feature. Notice that the samples and the graph convolution operation in DGCN-based method both preserve the topological information of road network and explore the inherent spatial distribution pattern of the traffic flows. Other machine learning-based methods, however, highly likely to exclude the spatial information when converting the sub-spanning tree-based sample into the node information sequence, which makes the relevant these methods unable to learn the graph feature of the road network comprehensively.

6.2.3. Comparison with other graph convolutional network-based methods

We compare our method with one original graph convolutional network-based method and four state-of-the-art graph convolutional network-based methods, i.e., reg-GCN, the expended-GCN, GCNMR and DMVST c.f. (Kipf and Welling, 2017; Kejani et al., 2020; Yua et al., 2020).

At first, the comparison between distance-based input matrix and the input matrix we defined in Section 5.2 was given: DGCN and DGCN(Dis-Mat) in Fig. 9. It is clear that the newly defined matrix have a better performance because the matrix could represent the correlation-based matrices on graph more accurate, whereas distance-based matrices cannot describe the changes of dynamic traffic flow in the road network.

Besides, it is clear that the DGCN-based method achieves the best performance w.r.t. all evaluation indexes, see the red line in all figures. Compared with other graph convolutional network-based methods, one distinct feature of the DGCN-based method is that it is proposed to tackle the digraph-structured data rather than undirected graph-structured data. In this scenario, the defined traffic information matrix is not symmetric anymore, and the feature extraction is different from the existing one. As for the comparison of expanded-GCN, this model uses the digraph-structured data. However, the performance of it still slightly lower than our method. This mainly because that model does not consider the relationship of multiple types of road sections under the three-dimensional road network structure, but only focuses on the relationship between urban ground roads.

Notice that the utilization of digraph-based representation can better characterize the traffic flow distribution pattern than the undirected graph-based one, especially for the traffic pattern such as the tidal phenomena.

7. Conclusion

In this paper, we propose a novel machine learning-based framework for traffic congestion recognition. The framework is established by traffic big data analysis methods and graph convolutional network techniques, and is particularly suitable for the exploitation of large volume of vehicular data to assist the V2X-based traffic services. In this framework, We first establish a novel digraph-based representation of the hybrid urban traffic network, so that different kinds of traffic data such as the correlation of related traffic flows and the fundamental traffic variables can be integrated. Then we provide a Directed Graph Convolutional Neural Network (DGCN)-based learning model to achieve traffic state classification and congestion recognition, including a spanning tree generation technique to acquire samples from the original traffic data. In particular, to utilize the digraph-based samples, a new type of graph feature extraction method is introduced and the graph Fourier transform is defined accordingly. In turn, the graph features and node information jointly capture the traffic congestion features. Extensive experiments are conducted, and the comparative results with other methods demonstrate the advantages of the proposed framework.

CRediT authorship contribution statement

Xiao Han: Conceptualization, Software, Writing - original draft. **Guojiang Shen:** Methodology, Writing - review & editing. **Xi Yang:** Investigation, Writing - review & editing. **Xiangjie Kong:** Supervision, Validation, Writing - review & editing.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (62073295, 62072409), and also partially supported by Zhejiang Provincial Natural Science Foundation (LR21F020003, LY20F030018).

References

Bauza, R., Gozalvez, J., 2013. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. J. Network Comput. Appl. 36 (5), 1295–1307.
Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2014. Spectral networks and locally connected networks on graphs. In: Proceedings of International Conference on Learning Representations.
Chen, S., Varma, R., Sandryhaila, A., Kovačević, J., 2015. Discrete signal processing on graphs: Sampling theory. IEEE Trans. Signal Process. 63 (24), 6510–6523.
Chen, X., Zhang, S., Li, L., Li, L., 2019. Adaptive rolling smoothing with heterogeneous data for traffic state estimation and prediction. IEEE Trans. Intell. Transp. Syst. 20 (4), 1247–1258.
Dai, H., Kozareva, Z., Dai, B., Smola, A., Song, L., 2018. Learning steady-states of iterative algorithms over graphs. In: Proceedings of the International Conference on Machine Learning, pp. 1114–1122.
Feng, X., Ling, X., Zheng, H., Chen, Z., X, Y., 2018. Adaptive multi-kernel SVM with spatial?temporal correlation for short-term traffic flow prediction. IEEE Trans. Intell. Transp. Syst. 20 (6), 2001–2013.
Gori, M., Monfardini, G., Scarselli, F., 2005. A new model for learning in graph domains. In: Proceedings of the International Joint Conference on Neural Networks, vol. 2, pp. 729–734.
Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs. In: Proceedings of Neural Information Processing Systems, pp. 1024–1034.
Henaff, M., Bruna, J., LeCun, Y., 2015. Deep convolutional networks on graph-structured data, arXiv:1506.05163.
Junghans, M., Leich, A., 2016. Traffic state estimation with Bayesian Networks at extremely low V2X penetration rates. In: Proceedings of the 19th International Conference on Information Fusion.
Kachroo, P., Agarwal, S., Piccoli, B., Özbay, K., 2017. Multiscale modeling and control architecture for V2X enabled traffic streams. IEEE Trans. Veh. Technol. 66 (6), 4616–4626.
Ke, J., Zheng, H., Yang, H., Chen, X.M., 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transport. Res. Part C: Emerging Technol. 85, 591–608.
Kejani, T., Dornaika, F., Talebi, H., 2020. Graph Convolution Networks with manifold regularization for semi-supervised learning. Neural Network 127, 160–167.
Kerner, Boris, 1999. Congested traffic flow: observations and theory. Transport. Res. Rec.: J. Transport. Res. Board 1678, 160–167.
Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks. In: Proceedings of International Conference of Learning Representations.
Kong, X., Liu, X., Jedari, B., Li, M., Wan, L., Xia, F., 2019. Mobile crowdsourcing in smart cities: technologies. Appl. Feature Challenges IEEE Internet Things J. 6 (5), 8095–8113.
Kong, X., Xia, F., Li, J., Hou, M., Li, M., Xiang, Y., 2020. A shared bus profiling scheme for smart cities based on heterogenrous mobile crowdsourced data. IEEE Trans. Industr. Inf. 16 (2), 1436–1444.
Li, Y., Xu, Y., Liu, Z., Hou, H., Zheng, Y., Xin, Y., Zhao, Y., 2020. Robust detection for network intrusion of industrial IoT based on multi-CNN fusion. Measurement 154 article no. 107450.

Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R., 2015. Gated graph sequence neural networks, arXiv:1511.05493.
Li, Y., Yu, R., Shahabi, C., Liu, Y., 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, arXiv preprint arXiv:1707.01926.
Micheli, A., 2009. Neural network for graphs: A contextual constructive approach. IEEE Trans. Neural Networks 20 (3), 498–511.
Néstor, C., Aquino-Santos, R., et al., 2016. Traffic congestion detection system through connected vehicles and big data. Sensors 16 (5) article no.599.
Niepert, M., Ahmed, M., Kutzkov, K., 2016. Learning convolutional neural networks for graphs. In: Proceedings of the International Conference on Machine Learning, pp. 2014–2023.
Qi, Y., Sherif, I., 2014. A Hidden Markov Model for short term prediction of traffic conditions on freeways. Transport. Res. Part C: Emerg. Technol. 43, 95–111.
Sandryhaila, A., Moura, J.M., 2013. Discrete signal processing on graphs. IEEE Trans. Signal Process. 61 (7), 1644–1656.
Sardellitti, S., Barbarossa, S., Di Lorenzo, P., 2017. On the graph Fourier transform for directed graphs. IEEE J. Sel. Top. Signal Process. 11 (6), 796–811.
Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2009. The graph neural network model. IEEE Trans. Neural Networks 20 (1), 61–80.
Shafipour, R., Khodabakhsh, A., Mateos, G., Nikolova, E., 2019. A directed graph Fourier transform with spread frequency components. IEEE Trans. Signal Process. 67 (4), 946–960.
Shen, G., Han, X., Zhou, J., Ruan, Z., Pan, Q., 2018. Research on intelligent analysis and depth fusion of multi-source traffic data. IEEE Access 6, 59329–59335.
Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P., 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30 (3), 83–98.
Singh, R., Chakraborty, A., Manoj, B., 2016. Graph Fourier transform based on directed Laplacian. In: Proceedings of International Conference on Signal Processing and Communications, pp. 1–6.
Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C., 2019. Graph wavenet for deep spatial-temporal graph modeling, arXiv preprint arXiv:1906.00121.
Xiong, K., Leng, S., He, J., Wu, F., Wang, Q., 2019. Recouping efficient safety distance in IoV-enhanced transportation systems. In: Proceedings of 2019 IEEE International Conference on Communications.
Yao, H., Wu, F., Ke, J., Tang, X. et al., 2018. Deep multi-view spatial-temporal network for taxi demand prediction. thirty-Second AAAI Conference on Artificial Intelligence.
Yua, B., Leeb, Y., Sohna, K., 2020. Forecasting road traffic speeds by considering area-wide spatiotemporal dependencies based on a graph convolutional neural network (GCN). Transport. Res. Part C: Emerg. Technol. 114, 189–204.
Yuan, C., Yu, X., Li, D., Xi, Y., 2019. Overall traffic mode prediction by VOMM approach and AR mining algorithm with large-scale data. IEEE Trans. Intell. Transp. Syst. 20 (4), 1508–1516.
Yu, B., Yin, H., Zhu, Z., 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 3634–3640.
Zhang, Y., He, Y., 2018. An advanced interactive-voting based map matching algorithm for low-sampling-rate GPS data. In: 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC), pp. 1–7.
Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., Li, T., 2018. Predicting citywide crowd flows using deep spatio-temporal residual networks. Artif. Intell. 259, 147–166.