A graph deep learning method for short-term traffic forecasting on large road networks

Yang Zhang

SpaceTimeLab, Department of Civil, Environmental and Geomatic Engineering, University College London, Gower Street, London WC1E 6BT, UK and College of Systems Engineering, National University of Defense Technology, Changsha 410073, China

&

Tao Cheng*

SpaceTimeLab, Department of Civil, Environmental and Geomatic Engineering, University College London, Gower Street, London WC1E 6BT, UK

&

Yibin Ren

SpaceTimeLab, Department of Civil, Environmental and Geomatic Engineering, University College London, Gower Street, London WC1E 6BT, UK and Qingdao Collaborative Innovation Centre of Marine Science and Technology, College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China

Abstract: *Short-term traffic flow prediction on a large-scale road network is challenging due to the complex spatial-temporal dependencies, the directed network topology and the high computational cost. To address the challenges, this article develops a graph deep learning framework to predict large-scale network traffic flow with high accuracy and efficiency. Specifically, we model the dynamics of the traffic flow on a road network as an irreducible and aperiodic Markov chain on a directed graph. Based on the representation, a novel spatial-temporal graph inception residual network (STGI-ResNet) is developed for network-based traffic prediction. This model integrates multiple spatial-temporal graph convolution (STGC) operators, residual learning and the inception structure. The proposed STGC operators can adaptively extract spatial-temporal features from multiple traffic periodicities while preserving the topology information of the road network. The proposed STGI-ResNet inherits the advantages of residual learning and inception structure to improve prediction accuracy, accelerate the model training process, and reduce difficult parameter tuning efforts. The computational complexity is linearly related to the number of road links, which enables city-wide short-term traffic prediction. Experiments using a car-hailing traffic dataset at 10, 30 and 60-minute intervals for a large road network in a Chinese city shows that the proposed model outperformed various state-of-the-art baselines for short-term network traffic flow prediction.*

1 INTRODUCTION

Short-term traffic flow forecasting aims to determine the traffic volume in the next time interval, usually in the range of five minutes to an hour (Do et al., 2018). It is one of the central components of intelligent transportation systems (ITSs). Practical applications of the accurate, reliable and real-time short-term traffic forecasts include incorporating the predictions into a traffic signal control scheme and enhancing overall road network management to reduce traffic congestion (Vlahogianni et al., 2014). It could also be used for route guidance and to decrease the request response time for car dispatching, saving time and money. In recent decades, traffic flow forecasting has attracted increasing research attention from the academic community.

Traditional traffic flow prediction falls into two categories: model-driven and data-driven. Model-driven approaches rely on the simulation of traffic flow dispersion and drivers' decision-making processes. Examples include the agent-based simulation (Manley et al., 2014) and the dynamic traffic assignment (DTA) model (Chiu et al., 2011; Hashemi and Abdelghany, 2015). These models can successfully reproduce real traffic situations and capture the complexity of a traffic network. However, model-driven approaches usually depend on prior knowledge of the environment (Abadi et al., 2015). Additionally, a simulation system is not easily transferable to other locations, due to the physical representation of the complex road network (Manley et al., 2014). Data-driven approaches have become popular due to the increasing availability of urban traffic data from various sensors, such as loop detectors, GPS-equipped devices and automatic traffic counters. Data-driven traffic flow forecasting can be further categorised into parametric approaches and nonparametric approaches.

Statistical parametric techniques are based on time-series methods. One of the most popular parametric models is the autoregressive integrated moving average (ARIMA) model (Box and Pierce, 1970; Shu et al., 2003), as well as its diverse variants, such as ARIMAX (Williams, 2001), seasonal ARIMA (Williams and Hoel, 2003), dynamic space-time ARIMA (Min et al., 2009) and local space-time ARIMA (Cheng et al., 2014). Other examples include Kalman filtering (Xie et al., 2007), the Bayesian network (Sun et al., 2006), the Markov Chain model (Yu et al., 2003) and wavelet methods (Jiang and Adeli, 2004; Adeli and Jiang, 2008). However, these methods were proposed for short-term traffic prediction in small urban arterial networks or freeways, and all suffer from high computational complexity.

Alternatively, a large number of nonparametric models have been developed to perform traffic prediction tasks. Such approaches do not make strong assumptions about the statistical structure of the traffic data, but instead, automatically learn the relationship between the inputs and outputs (Smith and Oswald, 2003). Commonly used nonparametric models include artificial neural networks (ANN) (Dharia and Adeli, 2003; Jiang and Adeli, 2005; Vlahogianni et al., 2007; Boto-Giralda et al., 2010), support vector regression (SVR) (Haworth et al., 2014) and fuzzy rule-based algorithms (Stathopoulos et al., 2008). To incorporate spatial correlation of the roads, spatial-temporal nonparametric models have been proposed. For example, Cai et al. (2016) utilised a k-nearest neighbour model based on a spatiotemporal state matrix to predict traffic states. Unfortunately, the shallow architectures of these traditional nonparametric models limited the forecasting accuracy due to nonlinear temporal dynamics and the complex spatial dependencies of traffic flow (Lv et al., 2015).

Recently, deep learning (DL), an advanced development of traditional machine learning methods, has attracted significant research interest. It has achieved great success in many areas, such as image classification and natural language processing (LeCun et al., 2015; He et al., 2016). The "deep" structure is created by stacking multiple layers. Thus, latent features are automatically extracted to model underlying, complicated, and nonlinear relationships in the data. In transportation research area, DL has been successfully applied to tackle many problems (Hashemi and Abdelghany, 2018; Nabian and Meidani, 2018; Zhang and Cheng, 2019). Concerning traffic flow prediction, Lv et al. (2015) first utilised a stacked autoencoder DL model to predict short-term road traffic flow and showed that the proposed model outperformed many standard machine learning models. However, the topology of the road network was not considered, such that the spatial dependencies between road segments were neglected. To learn the spatial-temporal characteristics of traffic flow, Zhang et al. (2016) proposed a DL model based on a convolution neural network (CNN). The research area was first divided into two-dimensional grids, just like an image with pixels. A temporal sequence was then input to model temporal dependencies. Convolution operators were used to learn the local spatial relationships between adjacent grids. Inspired by this work, researchers combined the CNN with the recurrent neural network (RNN) or long-short-term memory (LSTM) for grid-based spatial-temporal traffic prediction (Wu and Tan, 2016; Polson and Sokolov, 2017; Yu et al., 2017; Zhao et al., 2017). However, the abovementioned approaches all predicted traffic flow based on grid units, because classical CNNs can only capture spatial dependencies in a Euclid domain. Although these methods could easily generate a citywide traffic flow forecast, applications to many real-world scenarios are not practical.

Many researchers have claimed that the configuration of a city's road network plays an important role in traffic flow prediction (Zou et al., 2010; Cheng et al., 2012; Ren et al., 2019). Very recently, several researchers generalised the traditional convolutional operators to graph-structured data, which is in a non-Euclid space. This allowed for the

forecasting of traffic flow on a large-scale road network using DL models. In the existing literature, there are two primary methods used to generalise convolutional filters to graph-structured data: either from the vertex domain (Bruna et al., 2013; Niepert et al., 2016) or the spectral domain (Defferrard et al., 2016). The former strategies construct locally connected regions by selecting a fixed-length sequence of neighbours for each node, and the regular convolution operations are then conducted in these regions. Vertex-domain methods are suitable for both directed and undirected graphs. However, the number of selected neighbours is fixed, which does not comply with the intrinsic topology of road networks. Alternatively, Bruna et al. (2013) introduced generalising a CNN by operating on the spectrum of the graph's Laplacian. To reduce the computational complexity, Defferrard et al. (2016) proposed a fast localised spectral filter on the graphs. Very recently, Yu et al. (2017) utilised such a model for multi-step road-network traffic flow prediction. A limitation of this work was the assumption of the undirected graph-structure of the road network, which misrepresented the real topology. In addition, the DL models usually require a time-consuming parameter tuning process to ensure its performance. A previous study (Lv et al., 2015) also reported that for traffic flow prediction with different time horizons (e.g. 15-min and 60-min traffic flow predictions), the optimal parameter settings of a DL model varied substantially due to the variation of the complex spatial-temporal dependencies.

In summary, there are several issues not addressed in the current literature. First, the complex topology of the directed road networks has not been adequately considered in DL models, especially on a large scale (Do et al., 2018). More accurate representation of a road network needs to be investigated. Second, diverse spatial-temporal dependencies have not been fully leveraged. For example, traffic flow exhibits different temporal periodicities, and the spatial dependency may vary across the network. Third, the scalability of deep structures for traffic flow prediction on a large-scale network has become a concern. A DL model needs a more efficient architecture to achieve reasonable training time and to simplify the parameter tuning process. These are key challenges for DL-based models being practically applied.

To overcome the abovementioned issues, this paper proposes a graph-based DL framework for short-term traffic flow forecasting on a large-scale directed road network. We first propose to represent a road network as a directed graph with nodes that are road segments and edges that indicate the adjacent relationships. The dynamics of the network traffic flow are then modelled as an irreducible and aperiodic Markov chain on the directed graph, and its transition probability matrix is utilised to determine the edge weights of the graph. Based on the graph representation, a spatial-temporal graph inception residual network (STGI-ResNet) is developed by integrating a novel spatial-temporal graph convolution (STGC) operator, the residual learning technique and the inception structure.

The major contributions of this work are as follows. (1) The proposed weighted directed graph representation method and the flow model effectively capture the complex topology of the network and the traffic similarity between road segments. (2) The proposed STGC operator extracts spatial-temporal features from graph-structured data with computational complexity linearly related to the number of road segments, which suited traffic prediction on large-scale road networks. (3) The inception residual structure makes the STGI-ResNet model robust, effective and efficient. (4) The approach has been evaluated using a large car-hailing traffic flow data in Chengdu, China for 10-min, 30-min, and 60-min traffic prediction to demonstrate the advantages of the proposed model compared with baseline models and variants.

The reminder of this paper is organised as follows: Section 2 briefly describes preliminary concepts. Section 3 introduces the proposed STGI-ResNet model. Section 4 presents the case study and explicit discussions. Finally, Section 5 summarises the conclusions and future research directions.

2 PRELIMINARIES

2.1 A road network as a directed well-behaved graph

In this paper, we formulate the traffic flow forecasting problem on directed graphs, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A, W)$, where $\mathcal{V}$ is a set of N nodes, $\mathcal{E}$ is the edge set, $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix, and $W \in \mathbb{R}^{N \times N}$ is the weight matrix.

The structure of a road network determines the topology of the directed graph. The nodes of the graph represent the road segments and the edges indicate the adjacent relationships. Note that a two-way road is treated as two separate segments with opposite traffic directions, which are represented by two different nodes in the graph. A one-way road is represented by a single node. The directed edges between nodes are determined by the following four rules:

(1) If vehicles can travel from road segment u to v via a junction, there exists a directed edge from vertex u to v. For example, in Figure 1 (a), the traffic flow on segment 7 can diffuse to the road segments 1 and 4, but not the other way around. Therefore, node 7 has two edges to nodes 1 and 4, respectively.

(2) For a no-through road, the nodes representing the segments with the opposite traffic direction are connected at the dead end, because vehicles need to exit through the same direction as the entrance. For instance, segments 12 and 13 are on the same no-through road in Figure 1 (a).

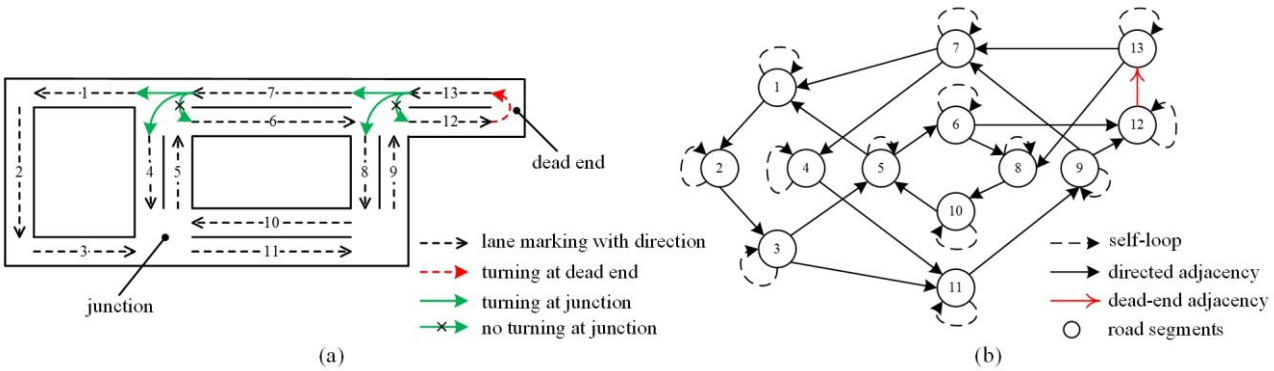

Figure 1 Representation of a road network as a directed well-behaved graph. (a) A real road network; (b) The corresponding directed graph.

There is then a directed edge from node 12 to 13 in Figure 1 (b), which indicates vehicles can turn around at the dead end and makes node 13 reachable from other nodes.

(3) For any roads, the nodes representing the segments on the same road but with opposite traffic directions are defined to be disconnected at junctions (not dead end), because it is unlikely that vehicles make U-turns at junctions. For example, there is no edge from node 13 to 12 or from node 7 to 6 in Figure 1 (b).

(4) A self-loop is also added to each node in Figure 1 (b) since each road is its own zeroth order neighbour, implying the future traffic flow of a road segment is affected by its own historical traffic data. In addition, all roads have different lengths and travel times. Adding self-loops takes into account different traffic remaining probabilities on the roads (Crisostomi et al., 2011).

The topology of the directed graph associated with the traffic road network has two important properties:

(1) Irreducibility: The adjacency matrix $A=(a_{ij})$ of a directed graph is irreducible if and only if the graph is strongly connected (Brualdi and Ryser, 1991). Any nonnegative matrix $B=(b_{ij})$ satisfying $b_{ij}\neq 0 \Leftrightarrow a_{ij}\neq 0$ is also irreducible, e.g., the corresponding edge weight matrix. Strong connectivity means every vertex is reachable from every other vertex in the graph. For any traffic road network, the corresponding directed graph representing the network must be strongly connected because it is possible to reach any road from any other road.

(2) Aperiodicity: A directed graph is said to be aperiodic if the greatest common divisor of the lengths of its cycles is one (Chung, 2005). Thus, a directed graph with one or more self-loops must be aperiodic.

In this paper, a strongly-connected and aperiodic graph is referred to as a 'well-behaved' graph. Based on the proposed rules, any road network can be abstracted as a directed 'well-behaved' graph, which has many favourable mathematical properties. These properties will be described and leveraged in the following sections.

2.2 Dynamic traffic flow as a Markov chain

The dynamics of traffic flow on a road network can be modelled as a Markov chain on a directed graph. In the Markov chain, the transition from one node of the graph to a successive node is defined as the amount of traffic flow directly diffusing from one road segment to another with a certain probability (Crisostomi et al., 2011; Anwar et al., 2015), which is characterised by a transition probability matrix (TPM).

The TPM in this paper is formulated as follows:

$$p_{uv}=\begin{cases}(tt_u-tt_{\min})/tt_u, & \text{if } u=v\\ (1-p_{uu})\cdot tp_{uv}, & \text{if } u\neq v \text{ and } (u,v) \text{ is an edge}\\ 0 & \text{otherwise.}\end{cases} \quad (1)$$

where tt_u is the average travel time passing the u-th road, $tt_{\min}$ is the smallest travel time among all roads and tp_{uv} is the junction turning probability. The TPM satisfies $p_{uv}>0$ and $\sum_v p_{uv}=1$. The diagonal element $p_{uu}\in[0,1)$ indicates the probability of the self-loop added to the u-th road, termed the traffic remaining probability. p_{uu} is computed by taking into account different travel times, which generally depend on the length of the road and the average travel speed. A longer road length and a slower travel speed indicate a higher remaining probability. In this way, the matrix P can be obtained after estimating the average travel time and junction turning probabilities from the collected historical data.

The TPM can serve as the weight matrix of the graph, that is $W=P$. A large edge weight indicates a high spatial similarity between two nodes. Using TPM, if the turning probability from road segment u to v is higher than from u to q, namely the edge weights $w_{uv}>w_{uq}$, it indicates the spatial similarity between u and v is higher than that between u and q. The turning probability from a road to downstream links with a large capacity is generally higher than to downstream links with a small capacity. In this way, the roadway capacity can be considered in the model.

The reasons to employ the Markov chain are twofold. First, the Markov chain can depict the collective transition behaviour of large aggregates of vehicles at a macroscopic level (not the exact behaviour of a single vehicle). Second, the TPM of the Markov chain can provide useful properties of the network topology and the weight pattern of the graph. For example, the TPM can be used to identify the most significant roads and the roads that are most congested (Crisostomi et al., 2011).

2.3 Problem formulation for traffic flow prediction

Network-based traffic flow forecasting is a spatial-temporal data prediction task considering the graph topology $\mathcal{G}$. Assuming that at time interval *t*, the observed traffic feature is $\boldsymbol{X}_t \in \mathbb{R}^{N\times M}$ where *N* is the number of road segments and *M* is the number of observed features for each road segment. The prediction target $\boldsymbol{x}_t \in \mathbb{R}^N$ is a column vector of traffic flow on the *N* roads, referred to as a graph signal. The forecasting task is formulated to learn a function $\mathcal{H}(\cdot)$ to estimate the traffic flow in the next time step given the historical observations $\boldsymbol{X}_t$ and the graph $\mathcal{G}$, written as

$$\hat{\boldsymbol{x}}_t = \mathcal{H}(\boldsymbol{X}_t; \mathcal{G}) \tag{2}$$

3 METHODOLOGIES

3.1 Deep learning framework

Figure 2 presents the proposed deep learning framework for short-term traffic forecasting on a large-scale directed road network. First, considering the diverse temporal periodic trends of traffic flow, three different types of temporal features, short, medium, and long-term temporal features, are defined to be used as the inputs to the model. The proposed STGI-ResNet for traffic forecasting is composed of multiple spatial-temporal graph inception residual (STGI-Residual) units, each of which consists of a shortcut for identity mapping and an inception structure. The former is utilised to accelerate the learning process and the latter has several parallel spatial-temporal graph convolution operators, which capture the diverse spatial-temporal dependencies between each road segment and its nearby neighbours for traffic flow prediction. Each module of the framework is described below.

3.2 Temporal dependencies

Time series of traffic flow exhibits diverse periodicities, including hourly, daily, and weekly periodicities (Lippi et al., 2013; Wu and Tan, 2016). If directly feeding a DL model with a very long sequence of previous observations for long-term temporal dependency modelling, it may substantially slow down the training process, especially when learning the spatial and temporal features simultaneously in a single model (Zhang et al., 2016). Alternatively, selecting highly-dependent timestamps as the inputs can efficiently train the model without decreasing its performance (Ma et al., 2014; Wu and Tan, 2016). In this paper, we define three types of temporal features, referred to as the short-, medium-, and long-term temporal features, to model the various periodicities of traffic flow. Assuming that the prediction time is *t*, the three types of features are defined as follows:

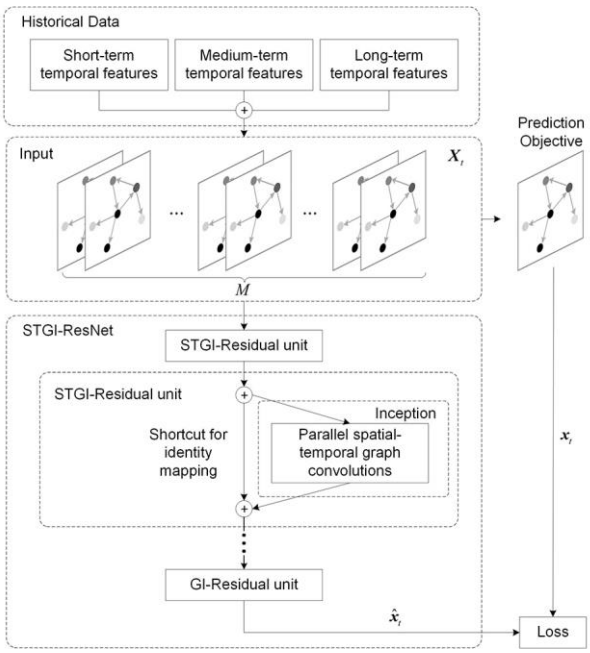

Figure 2 The proposed deep learning framework for short-term traffic flow prediction

Short-term temporal features: a series of historical traffic observations in the previous m_s time intervals, that is:

$$\boldsymbol{X}_t^s = [\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \cdots, \boldsymbol{x}_{t-m_s}] \in \mathbb{R}^{N\times m_s} \tag{3}$$

Medium-term temporal features: Assuming the number of time intervals of a medium period (e.g. daily periodicity) is t_m, a set of m_m medium-term temporal features is written as:

$$\boldsymbol{X}_t^m = [\boldsymbol{x}_{t-t_m\times 1}, \boldsymbol{x}_{t-t_m\times 2}, \cdots, \boldsymbol{x}_{t-t_m\times m_m}] \in \mathbb{R}^{N\times m_m} \tag{4}$$

Long-term temporal features: Assuming the number of time intervals of a long period (e.g. weekly or monthly periodicity) is t_l. A set of m_l long-term temporal features is written as:

$$\boldsymbol{X}_t^l = [\boldsymbol{x}_{t-t_l\times 1}, \boldsymbol{x}_{t-t_l\times 2}, \cdots, \boldsymbol{x}_{t-t_l\times m_l}] \in \mathbb{R}^{N\times m_l} \tag{5}$$

The three categories of temporal features are concatenated as the input for the proposed model, therefore the input graph signal in Eq. (2) is rewritten as

$$\boldsymbol{X}_t = \boldsymbol{X}_t^s \oplus \boldsymbol{X}_t^m \oplus \boldsymbol{X}_t^l \in \mathbb{R}^{N\times M} \tag{6}$$

where $M = m_s + m_m + m_l$ and $\oplus$ is the concatenation operator.

3.3 Spatial dependencies

Many existing works support the local spatial correlation and the heterogeneity of the traffic flow on road networks (May, 1990; Zou et al., 2010). To model the spatial dependencies on a directed network, a spectral graph convolution operator based on a directed Laplacian (Chung, 2005) is proposed.

3.3.1 Laplacian of Directed Graphs. In spectral graph theory, a graph's topology and the pattern of edge weights can be captured by its Laplacian matrix. For a weighted directed network, the Laplacian of directed graphs proposed by Chung (2005) is used. It is mathematically convenient for implementing the graph convolution on a directed graph.

According to graph theory, a Markov chain on a 'well-behaved' graph is irreducible and aperiodic, which converges to a unique stationary distribution ϕ (Chung, 2005) satisfying

$$\phi P = \phi, \text{ where } \sum\nolimits_v \phi(v) = 1 \tag{7}$$

ϕ is treated as a row vector. The Laplacian of a directed graph $\mathcal{G}$ is then defined by

$$\mathcal{L} = I_N - \frac{\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P^* \Phi^{1/2}}{2} \tag{8}$$

where I_N is an identity matrix, Φ is a diagonal matrix with $\Phi(v,v) = \phi(v)$ and P^* is the conjugate transpose of P. The symmetric Chung's Laplacian can be diagonalised as $\mathcal{L} = U \Lambda U^T$, where $\Lambda = \text{diag}([\lambda_0, \lambda_1, \cdots, \lambda_{N-1}]) \in \mathbb{R}^{N \times N}$ is the diagonal matrix of N real and non-negative eigenvalues (i.e., Laplacian spectrum) satisfying $0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{N-1}$ and $U = [u_0, \cdots u_{N-1}] \in \mathbb{R}^{N \times N}$ is the matrix of the associated orthonormal eigenvectors $\{u_i\}_{i=0}^{N-1}$ (i.e., graph Fourier basis) satisfying $UU^T = U^T U = I_N$. Based on these properties, spectral graph convolutional operations can be implemented on directed graphs.

3.3.2 Spectral Convolutions on Directed Graphs. Spectral convolution is a promising method used to capture the local spatial dependency of graph-structured data. The advantages of the operation are twofold. First, it can learn the similarity between nodes, considering both the topology and the edge weight pattern of the graph. Second, this operation can reduce the computational complexity from $\mathcal{O}(N^2)$ to a linear cost. This is significant for implementing traffic flow prediction on a large-scale road network with thousands of road segments.

Spectral graph convolution on a directed graph $\mathcal{G}$ with N nodes is defined as a graph signal $\boldsymbol{x} \in \mathbb{R}^N$ being filtered by a spectral graph filter g_θ in the Fourier domain, written as

$$\boldsymbol{y} = g_\theta *_{\mathcal{G}} \boldsymbol{x} = g_\theta(\mathcal{L})\boldsymbol{x} = U g_\theta(\Lambda) U^T \boldsymbol{x} \tag{9}$$

where $*_{\mathcal{G}}$ is the graph convolution operation on the graph $\mathcal{G}$, $\mathcal{L}$ is the graph Laplacian defined in Eq. (8), $U^T \boldsymbol{x} \in \mathbb{R}^N$ is the graph Fourier transformation of the signal $\boldsymbol{x}$ and $g_\theta(\Lambda)$ can be designed as a polynomial filter parameterised by $\boldsymbol{\theta} = [\theta_0, \cdots \theta_{K-1}]^T \in \mathbb{R}^K$:

$$g_\theta(\Lambda) = \begin{pmatrix} \sum_{k=0}^{K-1} \theta_k \lambda_0^{\,k} & & \\ & \ddots & \\ & & \sum_{k=0}^{K-1} \theta_k \lambda_{N-1}^{\,k} \end{pmatrix} = \sum_{k=0}^{K-1} \theta_k \Lambda^k \tag{10}$$

where $\boldsymbol{\theta}$ is learnable in the model and shared by all nodes on the graph, and K is referred to as the filter size hereinafter. The Eq. (9) can be rewritten as

$$\boldsymbol{y} = U \sum_{k=0}^{K-1} \theta_k \Lambda^k U^T \boldsymbol{x} = \sum_{k=0}^{K-1} \theta_k U \Lambda^k U^T \boldsymbol{x} = \sum_{k=0}^{K-1} \theta_k \mathcal{L}^k \boldsymbol{x} \tag{11}$$

According to spectral graph theory, the shortest path distance between vertices u and v is longer than K, such that $\mathcal{L}^K(u,v) = 0$. Therefore, a spectral graph filter of filter size K has access to nodes at most at K-1 hops. It means that the spectral graph convolution operation captures the spatial dependency between each road segment and its ith-order ($0 \le i \le K$) adjacent neighbours. As the Laplacian matrix is symmetric, from a traffic engineering point of view, the proposed spectral convolution can model the spatial dependency by explicitly collecting the traffic data on the upstream and downstream links for each road and then conduct convolution on these neighbours.

However, the computation complexity of $\mathcal{L}^k$ is high with $\mathcal{O}(N^2)$ due to the multiplication. To overcome this problem, the truncated Chebyshev polynomial is used to approximate $\mathcal{L}^k$ recurrently. The Chebyshev polynomial $T_k(x)$ of order k can be computed by the stable relation $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0(x) = 0$ and $T_1(x) = x$. Hammond et al. (2011) showed that for $x \in [-1,1]$, the stable recurrence with $T_k(x)$ bounded between -1 and 1 was ensured. Thus, the Laplacian is scaled as $\tilde{\mathcal{L}} = 2\mathcal{L} / \lambda_{N-1} - I_N$ so that the eigenvalues of $\tilde{\mathcal{L}}$ lie in [-1, 1]. The filtering operation is then written as

$$\boldsymbol{y} = g_\theta *_{\mathcal{G}} \boldsymbol{x} = g_\theta(\mathcal{L})\boldsymbol{x} = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathcal{L}})\boldsymbol{x} \in \mathbb{R}^N \tag{12}$$

where $T_k(\tilde{\mathcal{L}}) = 2\tilde{\mathcal{L}} T_{k-1}(\tilde{\mathcal{L}}) - T_{k-2}(\tilde{\mathcal{L}})$ with $T_0(\tilde{\mathcal{L}}) = 0$ and $T_1(\tilde{\mathcal{L}}) = \tilde{\mathcal{L}}$. Note that leveraging the sparse matrix multiplication technique, the computational complexity of $\tilde{\mathcal{L}}\boldsymbol{x}$ is only of $\mathcal{O}(K|\mathcal{E}|)$. The recurrent operation is thus of $\mathcal{O}(K|\mathcal{E}|)$ complexity.

3.4 Spatial-Temporal graph convolution (STGC)

To model the spatial-temporal dependency of the traffic flow, a spatial-temporal graph convolution (STGC) operator is developed by adaptively fusing the various temporal features and capturing spatial dependency simultaneously.

Figure 3 displays the proposed STGC operator with a spectral graph filter of filter size *K*=2. In this operator, to capture temporal dependency, *M* temporal features are adaptively summed up by *M* learnable parameters. To extract spatial features, the weighted sum is then fed into the proposed spectral graph filter. Finally, the rectified linear unit (ReLU), defined as $f(x)=\max(x,0)$, is used as the activation function to introduce nonlinearity into the model. ReLU is essentially a piecewise linear function, which is widely used in DL models. Empirical evidence suggests ReLU can accelerate the convergence of the model and it is less computation expensive (Krizhevsky et al., 2012). The proposed STGC operator is formulated as:

$$\boldsymbol{Y}=f\left(g_\theta *_{\mathcal{G}}\left(\boldsymbol{X}\boldsymbol{\omega}\right)+b\right)\in\mathbb{R}^N \tag{13}$$

where $\boldsymbol{X}\in\mathbb{R}^{N\times M}$ is the concatenation of *M* temporal features of all *N* graph nodes, $\boldsymbol{\omega}=(\omega_1,\cdots,\omega_M)\in\mathbb{R}^M$ is a row vector of *M* learnable parameters optimized during model training, $b\in\mathbb{R}$ is the bias and *f* is the ReLU function. The bias value allows for the shifting of the mapping function, which can be critical for successful learning.

The computational complexity of the proposed STGC operator is $\mathcal{O}(K|\mathcal{E}|+M)$, which is linearly related to the number of the road segments $|\mathcal{E}|$. Therefore, it is suitable for traffic flow prediction on a large-scale network.

3.5 Spatial-temporal graph inception residual networks

This section elaborates on the architecture of the proposed STGI-ResNet model, which combines a residual learning technique and an inception structure. The proposed STGC operator is integrated into the STGI-ResNet model for spatial-temporal modelling.

3.5.1 Residual structure. DL models always require a long time for model training. This paper proposes to utilise a residual learning structure to accelerate the convergence of the DL model. Residual learning was introduced by He et al. (2016). The core idea is to add an identity shortcut connection to preserve the inputs by skipping one or more layers. It is based on the hypothesis that learning a residual mapping is easier than directly fitting the desired underlying mapping without a reference. Suppose $\mathcal{H}(x)$ is the desired mapping, which uses an input *x* to predict the target *y*, i.e., $y=\mathcal{H}(x)$. As shown in Figure 4, rather than directly learning $\mathcal{H}(x)$ via a plain structure, a residual

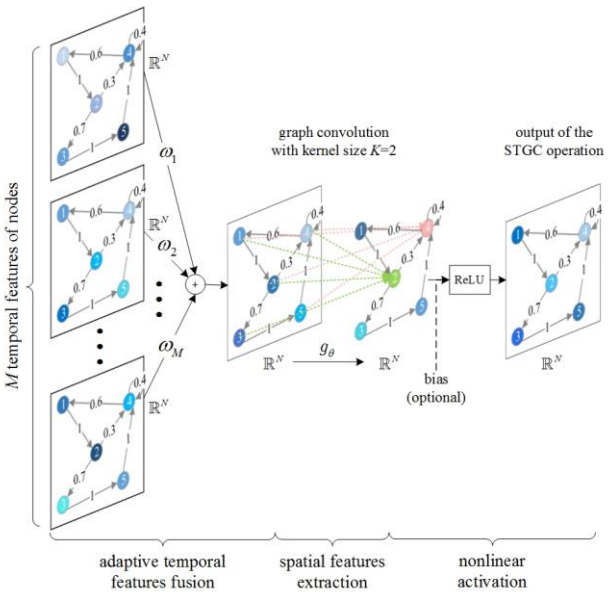

Figure 3 Spatial-temporal graph convolution operator with filter size *K*=2 (i.e., considering upstream and downstream neighbours at almost *K*-1 hops) for spatial-temporal feature extraction.

structure aims to fit as the residual part $\mathcal{F}(x)=\mathcal{H}(x)-x$. The original mapping is then recast into:

$$y=\mathcal{H}(x)=\mathcal{F}(x)+x \tag{14}$$

where the additional *x* is called an identity mapping. Practical applications have shown that with a residual structure, a network converges faster compared to its plain counterpart (He et al., 2016).

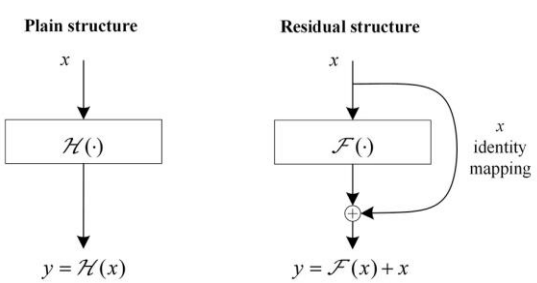

Figure 4 A residual learning structure vs. a plain counter part of it.

If *x* and $\mathcal{F}(x)$ in Eq. (14). are not the same dimensions, we can use a linear projection to *x* to match the dimensions. For example, suppose $x\in\mathbb{R}^{D_1\times D_2}$, *y* and $\mathcal{F}(\cdot)\in\mathbb{R}^{D_1\times 1}$. A linear projection is formulated as $y=\mathcal{F}(x)+xW_x$, where $W_x\in\mathbb{R}^{D_2\times 1}$ is for dimension matching.

3.5.2 Inception Structure. The inception network was first proposed by Szegedy et al. (2015) for image

classification using CNN. The 'inception' means making a network get 'wider' rather than 'deeper'. Traditionally, a DL model needs to stack many layers to achieve adequate performance. For example, the left network in Figure 5 has three sequential layers consisting of multiple STGC operators. A very deep configuration is computationally expensive, especially when increasing the filter size *K*. Alternatively, the 'inception' structure makes the network 'wider' by combing multiple layers with different filter size *K* in parallel on the same level. For example, the right inception architecture in Figure 5 employs three STGC layers parallelly to extract spatial-temporal features from different spatial spans. Empirical analysis proves that the inception structure can significantly improve the performance regarding computational efficiency and accuracy (Szegedy et al., 2015).

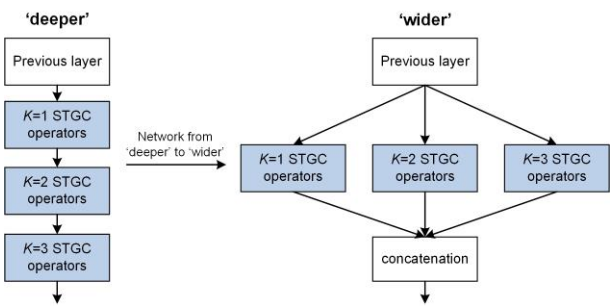

Figure 5 The concept of the **'**inception**'**: making a network get 'wider' rather than 'deeper'.

3.5.3 STGI-ResNet model. For traffic flow prediction on a large network, this study develops a novel STGI-ResNet model, which integrates the proposed spatial-temporal graph convolution, the residual learning technique and the inception structure. The STGI-ResNet model is a network consisting of *L* stacked STGI-Residual units, as illustrated in Figure 6.

Each STGI-Residual unit consists of a shortcut path and an inception structure. For the *l*-th ($1 \le l \le L$) STGI-Residual unit, there are three parallel STGC layers (visualised as blue blocks in Figure 6). Each STGC layer consists of N_l STGC operators. The filter size of the STGC operators in the three STGC layers is *K*=1, 2, and 3, respectively.

Suppose that in the *l*-th STGI-Residual unit, the output of the *i*-th ($i = 1, \cdots, N_l$) STGC operator in the STGC layer with filter size *K* is denoted as $\boldsymbol{Y}_{K,i}^{l} \in \mathbb{R}^{N}$. According to Eq. (13), given a graph $\mathcal{G}$, $\boldsymbol{Y}_{K,i}^{l}$ can be written as:

$$\boldsymbol{Y}_{K,i}^{l} = f\left(g_{\theta K,i}^{l} *_{\mathcal{G}} \left(\boldsymbol{X}^{l-1}\boldsymbol{\omega}_{K,i}^{l}\right) + b_{K,i}^{l}\right) \in \mathbb{R}^{N} \qquad (15)$$

where $\boldsymbol{X}^{l-1} \in \mathbb{R}^{N \times N_{l-1}}$ is the output of the (*l*-1)-th STGI-Residual unit, f is the ReLU function, and $g_{\theta K,i}^{l}$, $\boldsymbol{\omega}_{K,i}^{l}$, and $b_{K,i}^{l}$ are the parameters in the STGC operator. Thus, the *l*-th STGI-Residual unit can be formulated as:

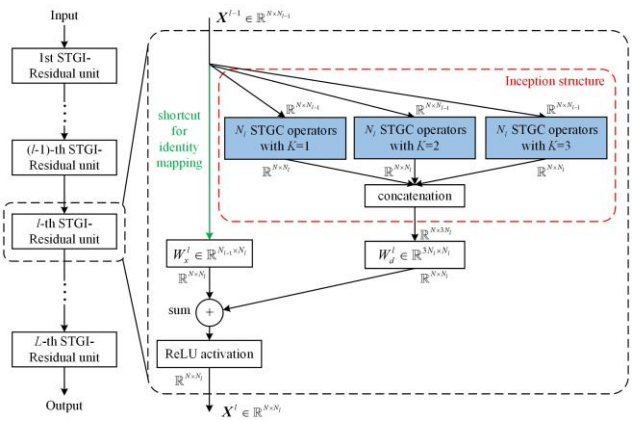

Figure 6 The STGI-ResNet model and the inner structure of a STGI-Residual unit. The three blue blocks are referred to as STGC layers.

$$\boldsymbol{X}^{l} = \mathcal{H}^{l}(\boldsymbol{X}^{l-1}; \mathcal{G}) = f\left(\boldsymbol{X}^{l-1}W_{x}^{l} + \boldsymbol{Y}_{concat}^{l}W_{d}^{l}\right) \qquad (16)$$

where $\boldsymbol{X}^{l} \in \mathbb{R}^{N \times N_l}$ is the output of the *l*-th STGI-Residual unit, $\boldsymbol{Y}_{concat}^{l} = \boldsymbol{Y}_{1,1}^{l} \oplus \cdots \oplus \boldsymbol{Y}_{1,N_l}^{l} \oplus \cdots \oplus \boldsymbol{Y}_{2,i}^{l} \oplus \cdots \oplus \boldsymbol{Y}_{3,N_l}^{l}$ is the concatenation of the outputs of the three parallel STGC layers, f is the ReLU function, and $W_{x}^{l} \in \mathbb{R}^{N_{l-1} \times N_l}$ and $W_{d}^{l} \in \mathbb{R}^{3N_l \times N_l}$ are used for dimension matching.

Concretely, in the *l*-th STGI-Residual unit, $\boldsymbol{X}^{l-1}W_{x}^{l}$ is the identity mapping of the output $\boldsymbol{X}^{l-1}$ from the previous unit via the shortcut path. The inception structure consisting of three STGC layers learns the residual part. It can fuse various temporal features and extract spatial dependencies from different spatial spans. To introduce the residual connection, the two matrices W_{x}^{l} and W_{d}^{l} are employed to match the dimensions of the outputs of the identity mapping and the inception structure, respectively. Empirical evidence suggests that the matrix W_{x}^{l} in residual learning can be just set as an all-ones matrix (He et al., 2016). On the contrary, W_{d}^{l} is a learnable parameter matrix, which is equivalent to the ensemble of different spatial-temporal features. For a more direct and concrete understanding of the dimension transformation process, the dimensions of the input and output of each STGC layer in the STGI-Residual units are demonstrated in Figure 6.

Overall, the proposed STGI-ResNet model is a stack of *L* STGI-Residual units. At prediction time point *t*, the estimated short-term traffic flow $\hat{\boldsymbol{x}}_t$ using STGI-ResNet can be formulated by rewriting Eq. (2) as:

$$\hat{\boldsymbol{x}}_t = \mathcal{H}(\boldsymbol{X}_t; \mathcal{G}) = \mathcal{H}^{L} \cdots \mathcal{H}^{2}\mathcal{H}^{1}(\boldsymbol{X}_t; \mathcal{G}) \qquad (17)$$

where $\boldsymbol{X}_t$ is defined in Eq. (6) and $\mathcal{H}^l (1 \le l \le L-1)$ is defined in Eq. (16). Note that the last STGI-Residual unit $\mathcal{H}^L = \boldsymbol{X}^{L-1} W_x^L + \boldsymbol{Y}_{concat}^L W_d^L$ does not have the ReLU activation function before the final outputs, allowing for those outputs to be negative or positive.

In traffic flow prediction, the 'inception' architecture is adopted because the spatial dependencies may change when using different time intervals for traffic forecasting. In addition, the spatial dependency might vary across the traffic network. The heterogeneity of the spatial-temporal dependency makes it difficult to choose the appropriate filter size *K* for graph convolution in different traffic forecasting tasks. When applying STGC operators in parallel, the model automatically extracts different spatial-temporal features. Thus, the optimal *K* does not need to be separately specified for different traffic flow prediction tasks, which saves on parameter-tuning efforts to a large extent. Furthermore, the residual connection is applied with the same purpose of accelerating the training process for real-time traffic flow prediction.

Although the network looks complicated, our model is in analogy with *N* dependent continuous piecewise linear (CPWL) functions, which share the parameters $\boldsymbol{\theta}$ in each graph filter $g_{\theta K,i}^l$. Thus, the *N* CPWL functions affect each other, constrain the parameters, and ensure the validity of multitask regression.

3.6 Parameter Learning

The proposed STGI-ResNet model can be trained to predict $\boldsymbol{x}_t$ from a sequence of historical observations $\boldsymbol{X}_t$ (defined in Eq. (2)) by minimizing the square error between the ground truths $\boldsymbol{x}_t$ and the prediction values $\hat{\boldsymbol{x}}_t$. The loss function is defined as:

$$loss = \left\| \boldsymbol{x}_t - \hat{\boldsymbol{x}}_t \right\|_2^2 \tag{18}$$

Table 1 shows the training process of the proposed model. All instances are divided into a training set, a validation set, and a testing set. The validation set is used for monitoring the change in predictive performance. The variables of the model are only saved when its performance increases, in order to prevent overfitting. The validation and testing datasets play no role in optimising the model's parameters. The final performance evaluation of STGI-ResNet is conducted on the test set only.

In the training process, all learnable parameters are randomly initialized. After that, the training set is iteratively fed into the model in batches. For each batch, the total loss is calculated. All learnable parameters are then adjusted via back-propagation and Adam optimizer (Kingma and Ba, 2014). After meeting the convergence criteria on the validation dataset, all learnable parameters can be learned. When using the model, we do not need to adjust any parameters other than the inputs.

Table 1
The parameter training process of STGI-ResNet

Algorithm 1: STGI-ResNet Training Algorithm

Input: historical observations: $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_{n-1}\}$;
the number of time intervals in medium and long periods: t_m and t_l ;
the number of short-, medium-, and long-term features: m_s , m_m , m_l ;
the graph $\mathcal{G}$;

Output: STGI-ResNet with well-trained parameters $\mathcal{W}_\theta$

// construct a set of input-output instances $\mathcal{D}$
$\mathcal{D} \leftarrow \varnothing$
for available time interval *t* **do**
get temporal features $\boldsymbol{X}_t^s$, $\boldsymbol{X}_t^m$, $\boldsymbol{X}_t^l$ using Eqs. (3-5)
$\boldsymbol{X}_t = \boldsymbol{X}_t^s \oplus \boldsymbol{X}_t^m \oplus \boldsymbol{X}_t^l$
// $\boldsymbol{x}_t$ *is the prediction target at time t*
put a training instance ($\boldsymbol{X}_t$, $\boldsymbol{x}_t$) into $\mathcal{D}$
end for
divide $\mathcal{D}$ into training, validation and testing datasets $\mathcal{D}_{train}$, $\mathcal{D}_{val}$, $\mathcal{D}_{test}$
// *training STGI-ResNet model*
initialise all learnable parameters in STGI-ResNet
repeat
randomly select a batch of instances $\mathcal{D}_b$ from $\mathcal{D}_{train}$
optimize $\mathcal{W}_\theta$ by minimizing the loss function
until convergence criteria is met in $\mathcal{D}_{val}$

4 EXPERIMENTS

In this section, an empirical study is provided to validate the effectiveness of the proposed STGI-ResNet model. All experiments were implemented on a computer workstation with 40 CPU cores (Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz (two processors)), 256 GB RAM and one GPU (NVIDIA Quadro K2200). Details are provided below.

4.1 Data source and pre-processing

The data used in this case study are from an open source shared by the Didi GAIA Initiative, which provides researchers with access to real-life, high-quality anonymised data for academic purposes. The dataset contains over 100GB of 30-day route data from the 1st to the 30th of November 2016, collected via Didi's smartphone app, depicting the dynamic car-hailing traffic flow on a road network. There were no extreme weather conditions or special days during this period. Each record in the data includes the anonymised driver ID, order ID, longitude, latitude, and time stamp with an accuracy of 2-4s. The study area covers the northeast (from 104.129591°

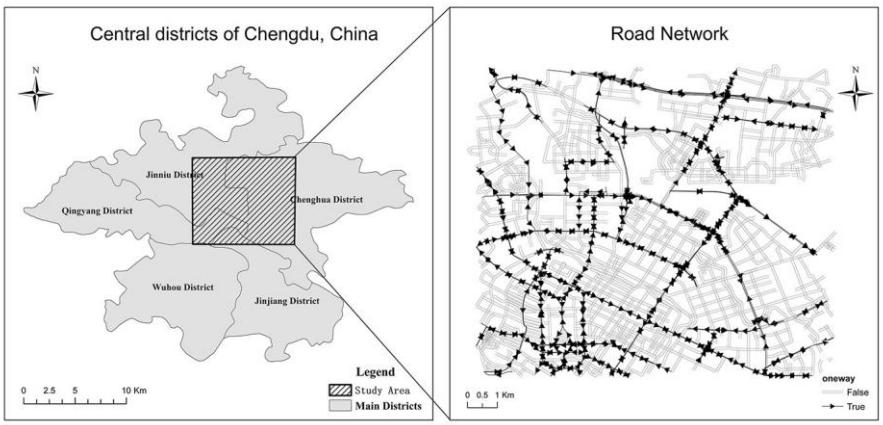

Figure 7 Road network of the research area.

E to 104.042102° E in longitude and from 30.727818° N to 30.652828° N in latitude) of Chengdu, which is the capital of China's Sichuan province. The road network was downloaded from OpenStreetMap on the 8th January 2018 using the Python package OSMnx (Boeing, 2017). It contains total of 2616 road segments consisting of 1021 one-way roads and 1595 two-way roads (as shown Figure 7). All two-way roads are treated as two segments with opposite directions.

To obtain the traffic flow data, the Hidden Markov map matching algorithm (Newson and Krumm, 2009) is first used to match the route data to the road network. The data are then aggregated into 10-min, 30-min and 60-min time intervals. Every dataset is scaled to the range [0, 1] using the Min-Max scaling method (Pedregosa et al., 2011) to speed up gradient descent during model training. In this experiment, the medium and long-term temporal features are set to be the daily and weekly features, namely setting t_m and t_l in Eq. (4) and (5) to be the number of time intervals during one-day and one-week, respectively. Note that if longer-term traffic flow data were available, other periodicities could have been easily integrated into the input of the proposed model, such as seasonal and yearly trends.

The road network within the research area is only a part of the entire traffic network. Some road segments at the edge of this area are only reachable from other roads that were excluded from the study area. This may lead to a weakly connected graph. To ensure the irreducibility of the graph, its strongly connected component is first found using the Python package 'networkx' (Hagberg et al., 2008). In the end, the network in this study consisted of 4098 road segments. In addition, the average travel times of all road segments and the junction turning probabilities are estimated from the collected trajectory dataset. According to Eq. (1), the transition probability matrix P is obtained.

The road network is then represented as a weighted directed graph, as shown in Figure 8. The road network in Figure 8 (a) and the nodes in Figure 8 (c) are coloured by the value of closeness centrality, which is the average distance from a given starting node to all other nodes in the network (Heymann, 2014). Therefore, the darkness diffuses from the centre to the margins. The closeness centrality is only used to match the road network with the directed graph for convenience. Thus, a detailed discussion of closeness centrality is not provided. A small section of the road network is shown in Figure 8 (b) and the corresponding directed graph (Figure 8 (d)) displays the details of the representation. Taking road No. 2633 as an example, its downstream road segments are No. 1421, 1959, 2587, and 2595, which are represented as the successors in the directed graph. Meanwhile, road No. 2634 has a dead end, thus, road No. 2633 with the opposite traffic direction is treated as its downstream road and road segment 2634 is represented as a predecessor of node 2633.

4.2 Performance metrics

In this paper, four metrics are employed to evaluate the prediction performance, namely Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Normalised RMSE (NRMSE), and Mean Absolute Percentage Error (MAPE). Let $\{x_i^{(j)}\}_{i=1}^n$ and $\{x_i'^{(j)}\}_{i=1}^n$ denote a sequence of the actual and predicted traffic flow of the j-th road ($j \in [1, N]$), respectively. The definitions of the four metrics are written as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{nN}\sum_{j=1}^{N}\sum_{i=1}^{n}\left(x_i^{(j)} - x_i'^{(j)}\right)^2} \tag{19}$$

$$\mathrm{MAE} = \frac{1}{nN}\sum_{j=1}^{N}\sum_{i=1}^{n}\left|x_i^{(j)} - x_i'^{(j)}\right| \tag{20}$$

$$\mathrm{NRMSE} = \frac{1}{N}\sum_{j=1}^{N}\frac{\mathrm{RMSE}_j}{x_{\max}^{(j)} - x_{\min}^{(j)}} \times 100\% \tag{21}$$

$$\mathrm{MAPE} = \frac{1}{nN}\sum_{j=1}^{N}\sum_{i=1}^{n}\left|\frac{x_i^{(j)} - x_i'^{(j)}}{x_i^{(j)}}\right| \times 100\% \tag{22}$$

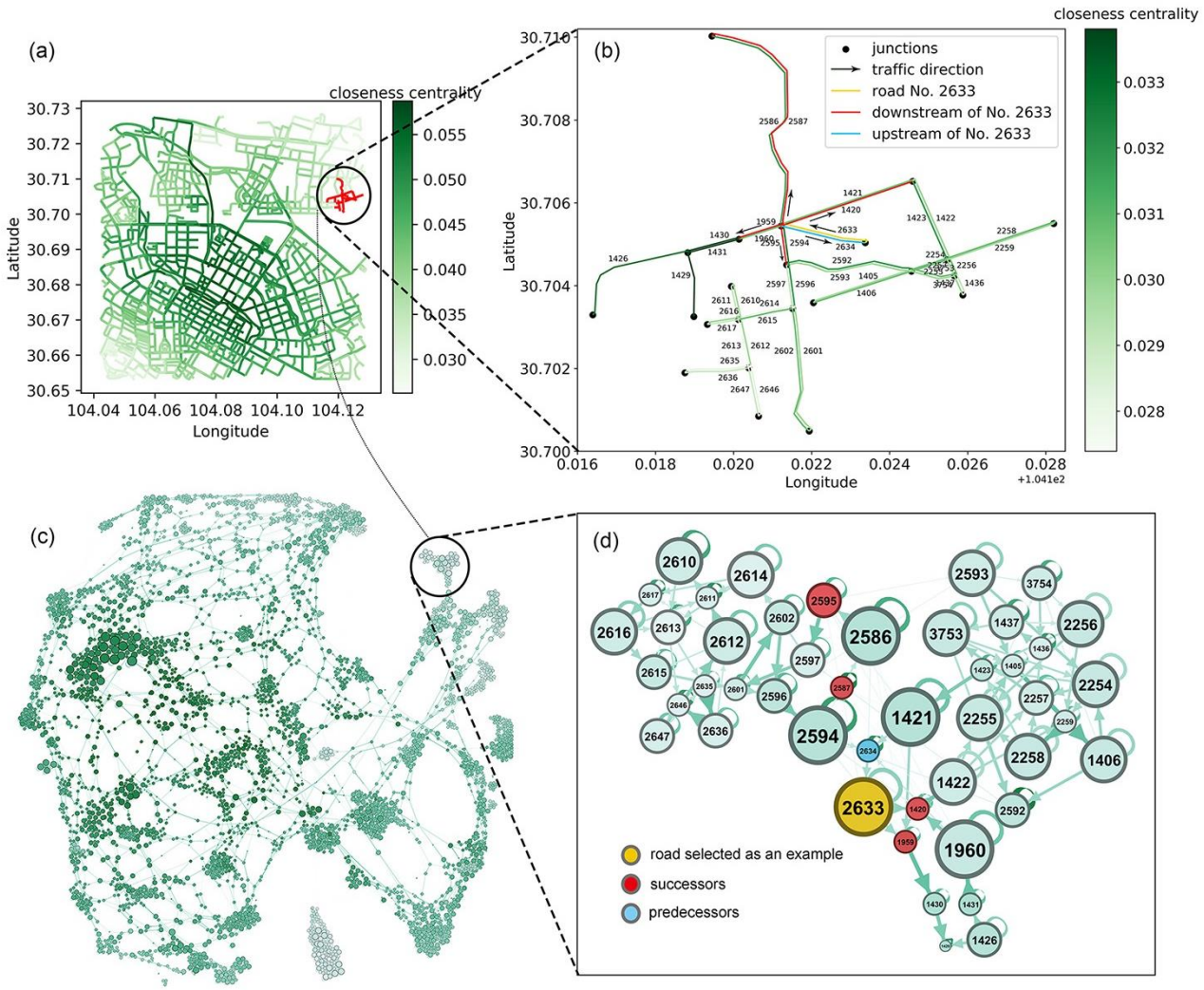

Figure 8 Representation of a road network as a weighted directed graph. Upper: the original road network. The darkness of the green colour is proportional to the closeness centrality of each node. Lower: the correspondent directed graph. The size of each node is proportional to its out-degree. The darker colour of nodes or edges implies larger closeness centrality or weights. A small section of the road network is enlarged in to show details. Road No. 2633 is selected as an example to visualise its upstream and downstream roads. Correspondingly, four successors and a predecessor of the node 2633 can be seen in the right lower plot.

where RMSE_j, $x_{\max}^{(j)}$, and $x_{\min}^{(j)}$ are the RMSE, maximum, and minimum traffic flow of the *j*-th road.

RMSE and MAE are used to measure the absolute errors between the ground truth and the prediction, while NRMSE and MAPE are relative error measurements used to remove the scale effect of different flow levels. The units of RMSE and MAE are the same as for the traffic flow; i.e., the number of vehicles aggregated during a time slot. Considering that the highly congested road segments should be paid more attention, MAPE@10 (i.e., MAPE in the road segments with the top 10% largest traffic flow) is taken to measure the model's predictive capability of the highly crowded roads.

4.3 Structure of STGI-ResNet

In this paper, the STGI-ResNet is used to forecast 10-min, 30-min, and 60-min traffic flows. The proposed framework contains several parameters that have to be defined to build the STGI-ResNet configuration. First, we need to determine the length of the inputs for short-, medium-, and long-term temporal sequences: m_s, m_m and m_l. Second, the number of STGI-Residual units have to be specified., as shown in Figure 6. Additionally, although the

Table 2
RMSE and training time of several selected STGI-ResNet configurations during the grid search procedure.

No.	m_s	Number of STGC operators	Number of units L	RMSE	Training time (s)
1	1	4	1	9.97	3.09
2	3	8	1	9.29	6.41
3	3	8	2	9.27	23.96
4	3	16	2	8.67	44.52
5	3	8	3	8.55	87.86
6	3	16	3	8.03	124.81
7	6	16	3	8.63	278.54
8	6	32	3	8.51	594.12

filter size K of each layer in the STGI-Residual unit is fixed, the number of STGC operators in each layer has to be determined.

Considering the available data is for one month, we let t_m and t_l in Eqs. (4) and (5) be the number of time intervals of one-day and one-week, respectively. This means to take into account the hourly, daily and weekly periodicities of traffic flow in the model. For simplicity, m_m and m_l are fixed to be one, as suggested in an existing work (Zhang et al., 2017). However, we do not claim the optimality of this setting. If longer time series data becomes available, further discussions should be conducted on the optimal setting of m_m and m_l. The number of short-term temporal features m_s is chosen from one to six. For simplicity, the number of STGC operators of the STGC layers in each STGI-Residual unit is set to be the same, chosen from [4, 8, 16, 32]. According to the literature (Lv et al., 2015), the number of layers in a neural network or DL model for traffic prediction should not be too small or too large. As one STGI-Residual unit consists of three STGC layers, the number of STGI-Residual units in STGI-ResNet is chosen from a small range of [1, 3].

As the previous one-week data are used to construct the input sequence, the total input-output pairs cover 23 days. Therefore, the first 17 days are used as the training set, the next two days are used as the validation set and the last four days are the testing set.

The STGI-ResNet model was implemented in TensorFlow 1.2 (Abadi et al., 2016), a Python-based deep learning library with GPU support. The proposed model was trained using the GPU. The number of training epochs is fixed to 100 (enough for convergence) and the batch size is 24. In the Adam optimizer, the learning rate is initialised at 0.01 and is then exponentially decayed every 50 steps with a base of 0.96. To determine the optimal configuration, the grid search method is used to find the best parameter settings by changing one of the parameters while keeping the others unchanged. The study starts with short hourly temporal features, shallow architectures and a low number of spectral graph filters. It gradually increases the number of hourly temporal features, STGI-Residual units, and filters to evaluate if the predictive performance increases. For simplicity, the 60-min traffic flow prediction is taken as an example and the prediction performance of several important configurations are given in Table 2. This highlights the influence of parameter settings in terms of RMSE and training time. The RMSE decreases as the number of short-term features, filters, and units increase. It declines to 8.03 in the No. 6 configuration but then slightly increases to 8.63 in the No. 7 configuration. The training time increases dramatically as the configuration becomes deeper and more complex. The training time of the No. 8 configuration is over four times that of the No. 6 configuration. Finally, the No.6 configuration is chosen as the optimal one for traffic forecasting.

4.4 Model comparison and analysis

4.4.1 Performance comparison. In this section, the proposed STGI-ResNet model is compared with several baselines, including historical average (HA), moving average (MA), ARIMA, SVR, LSTM, and graph convolution LSTM (GC-LSTM). Among all baselines, the HA is a simple model that forecasts the future traffic flow as equal to the average of all historical observations in the same time interval on the same day of the week. The MA model predicts the traffic using the average of the historical values in previous *m*-time steps. ARIMA is a widely used statistical model for traffic prediction. In addition, SVR, LSTM and GC-LSTM are three state-of-the-art machine/deep learning models. SVR and LSTM model the traffic flow as a time series and GC-LSTM takes into account spatial dependency. Similar to the parameter tuning procedure of the proposed model, the optimal parameter settings of SVR, LSTM and GC-LSTM are also determined via the grid search method (Pedregosa et al., 2011).

The field data used for model comparison are the traffic flows of discrete time series aggregated every 10-min, 30-min, and 60-min alongside the road network. In all cases, the data are split as described in Section 4.3 for training,

Table 3
Performance comparison between the proposed STGI-ResNet and various baselines.

Task	Metrics	HA	MA	ARIMA	SVR	LSTM	GC-LSTM	STGI-ResNet
10-min	RMSE	3.29	3.14	3.11	3.06	2.94	2.79	2.05
	MAE	1.81	1.78	1.75	1.79	1.68	1.51	1.41
	NRMSE	16.65%	15.37%	16.20%	25.07%	15.33%	14.43%	13.05%
	MAPE@10	32.99%	30.89%	27.99%	22.39%	28.49%	27.48%	22.18%
	Training time	-	-	8.45d	1848.5s	5135.2s	7045.28s	725.14s
30-min	RMSE	7.25	7.62	7.31	7.12	6.05	5.93	5.14
	MAE	3.56	3.59	3.48	4.08	3.66	3.01	2.50
	NRMSE	17.87%	17.81%	16.52%	22.93%	16.07%	14.78%	12.71%
	MAPE@10	21.72%	20.72%	20.11%	19.71%	17.71%	14.79%	11.35%
	Training time	-	-	3.52d	119.84s	1705.2s	2357.4s	241.10s
60-min	RMSE	12.87	19.05	15.25	14.28	10.35	9.49	8.03
	MAE	6.48	8.46	7.05	7.55	5.99	4.51	3.89
	NRMSE	16.54%	20.32%	17.30%	30.30%	15.22%	14.32%	12.51%
	MAPE@10	17.86%	24.65%	19.46%	21.82%	14.63%	11.98%	9.21%
	Training time	-	-	1.94d	39.8s	865.5s	1109.1s	124.8s

validation and testing. Table 3 shows the prediction errors calculated based on the testing data and the training time of each model.

In terms of prediction accuracy, GC-LSTM and the proposed STGI-ResNet model outperforms HA, MA, ARIMA, SVR, and LSTM for short-term traffic flow prediction. This validates that spatial and temporal features should be considered simultaneously in network-based short-term traffic prediction. In addition, STGI-ResNet performs better than GC-LSTM. This could be because LSTM cannot model very long-range temporal dependencies (Zhang et al., 2017). Another possible reason could be that STGI-ResNet can capture various spatial dependencies via the three parallel graph convolution operators with different filter sizes.

Comparing to the optimal baseline, the MAE of STGI-ResNet decreases by 6.62%, 16.9%, and 13.74% in the three prediction tasks. In terms of the average NRMSE, the prediction performance of 10-min, 30-min and 60-min traffic flow prediction is improved by 9.56%, 14.00%, and 12.64%, respectively. Additionally, STGI-ResNet is superior to other models in terms of MAPE@10, which indicates its strong capability to predict traffic flow on congested roads.

To compare operational efficiency, the training time of each model is listed in Table 3 (the runtime of the testing procedure is negligible). The training time of the three deep learning models is measured in the GPU environment. Among the first three statistical models, HA and MA do not require any time for training but perform poorly, while the ARIMA takes the longest time to fit its parameters because of the large number of measurements. Comparing machine/deep learning models, the training time of SVR is the shortest in the 30-min and 60-min traffic flow prediction but is twice as long as that of STGI-ResNet in the 10-min traffic flow prediction. On average, STGI-ResNet ranks second in terms of efficiency and its training time is much shorter than the LSTM-based models. The longest training time of STGI-ResNet is about 725 seconds, which is acceptable for practical applications.

To validate the performance difference between the baselines and the proposed STGI-ResNet, a two-sided paired t-test is conducted to determine if the mean values of two sets of prediction errors are significantly different. The null hypothesis for the two-sided t-test is that the mean prediction errors of STGI-ResNet and the baseline method have no difference, whereas the alternative hypothesis is significant difference. If a p-value is smaller than 0.025 (equivalent to a t-statistic larger than the corresponding critical value), then the null hypothesis of identical means could be rejected. In this case, the sample size (the number of road segments) is very large so that the critical value approximately equals to 1.96. The results of the paired t-test on NRMSE are taken as an example, listed in Table 4. It indicates that the mean NRMSE of the proposed model is significantly smaller than those of the baselines. However, the p-value of the paired t-test can only inform the statistical significance, not the quantitative magnitude of significance. And the large sample size may easily demonstrate a significant difference. To supplement the t-test, the effect size is employed as a measure of magnitude (Sullivan and Feinn, 2012), which is independent of the sample size. Cohen's term d is utilised as the effect size index. To interpret the values of the effect size, Cohen

Table 4
The paired t-test and effect size of NRMSEs of the baselines and proposed STGI-ResNet model

Task	Comparison	t-statistics	If significant	Cohen's d	Effect size
10min	HA vs. STGI-ResNet	88.64	Yes	1.01	large
	MA vs. STGI-ResNet	161.95	Yes	1.04	large
	ARIMA vs. STGI-ResNet	59.61	Yes	0.92	large
	SVM vs. STGI-ResNet	132.13	Yes	0.95	large
	LSTM vs. STGI-ResNet	65.95	Yes	1.02	large
	GC-LSTM vs. STGI-ResNet	43.13	Yes	0.88	large
30min	HA vs. STGI-ResNet	50.97	Yes	0.81	large
	MA vs. STGI-ResNet	118.95	Yes	0.86	large
	ARIMA vs. STGI-ResNet	61.31	Yes	0.79	medium
	SVM vs. STGI-ResNet	50.58	Yes	1.01	large
	LSTM vs. STGI-ResNet	65.71	Yes	0.88	large
	GC-LSTM vs. STGI-ResNet	90.61	Yes	0.89	large
60min	HA vs. STGI-ResNet	51.13	Yes	0.88	large
	MA vs. STGI-ResNet	203.45	Yes	2.14	large
	ARIMA vs. STGI-ResNet	66.07	Yes	0.98	large
	SVM vs. STGI-ResNet	53.96	Yes	1.09	large
	LSTM vs. STGI-ResNet	68.04	Yes	0.79	medium
	GC-LSTM vs. STGI-ResNet	94.11	Yes	0.77	medium

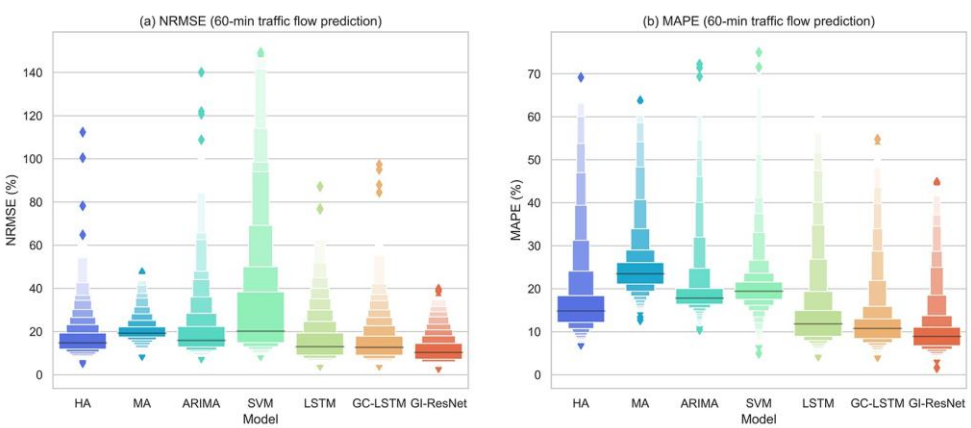

Figure 9 Letter-value plots of the NRMSE and MAPE of road segments for different models in 60-min traffic flow prediction. The black line displays the median value. The height of each box is fixed by the letter values and the width of each box is proportional to the percentage of data covered. The proportion of data believed to be outliers is 0.007.

(1988) has suggested using the following rule of thumb: small effect size (d=0.2), medium effect size (d=0.5), and large effect size ($d \geqslant 0.8$). The effect size values are listed in Table 4. Results show that most of the effect sizes are large. Although there are three cases with medium effect sizes, the Cohen's d values are very close to 0.8.

To show the comparison results in a more concrete way, the 60-min traffic prediction is taken as an example to show the letter-value plot (Hofmann et al., 2017) of prediction errors in Figure 9. It shows the distribution of NRMSE and MAPE for 4089 road segments across different models. The average relative errors of STGI-ResNet are smaller than other models. SVR and MA perform the worst in terms of the average NRMSE and MAPE, respectively. Regarding STGI-ResNet, its largest NRSME and MAPE values are 39.6% and 44.7%, respectively, which is notably lower than various baselines. In addition, its letter-value plots had lighter tails than the baselines, thus STGI-ResNet has fewer extreme outliers.

4.4.2 Prediction visualisation. Three road segments are taken as typical examples to demonstrate the 60-min traffic prediction results of different models in Figure 10. Similar visualisations can be obtained from 10-min and 30-min prediction results but are not provided here for conciseness.

Road No. 237 is the second North Section of the First Ring Road of Chengdu. This primary road has high traffic volume during non-sleeping hours and exhibited a recurrent traffic pattern during the test period. Road No. 3871 is a secondary road on the Third Ring Road from Fenghuang

Overpass towards Pengcheng Overpass. According to information published by 'chengdujiaojing', the official website of the Chengdu Traffic Management Bureau on Sina Weibo (a Chinese microblogging web), this road segment was congested from 10:00 to 11:00 am due to a traffic accident occurring at 9:47 am on its downstream road. The non-recurrent traffic peak could be observed within the dashed box in Figure 10 (b). Road No. 3777 is Wenmiao Back Street. It is a local street with relatively low traffic capacity, and its traffic pattern on weekends (27th Nov.) was significantly distinct from that on weekdays (28th-30th Nov.). In Figure 10, the upper plot in each subfigure displays the comparison between the ground truth and the predictions, and the lower plot presents the error derived from the ground truth for the seven different algorithms.

Overall, the proposed model can predict the correct traffic flow trend in heavy, medium, and low conditions. STGI-ResNet yields more accurate results than the other models. The prediction errors of STGI-ResNet are closest

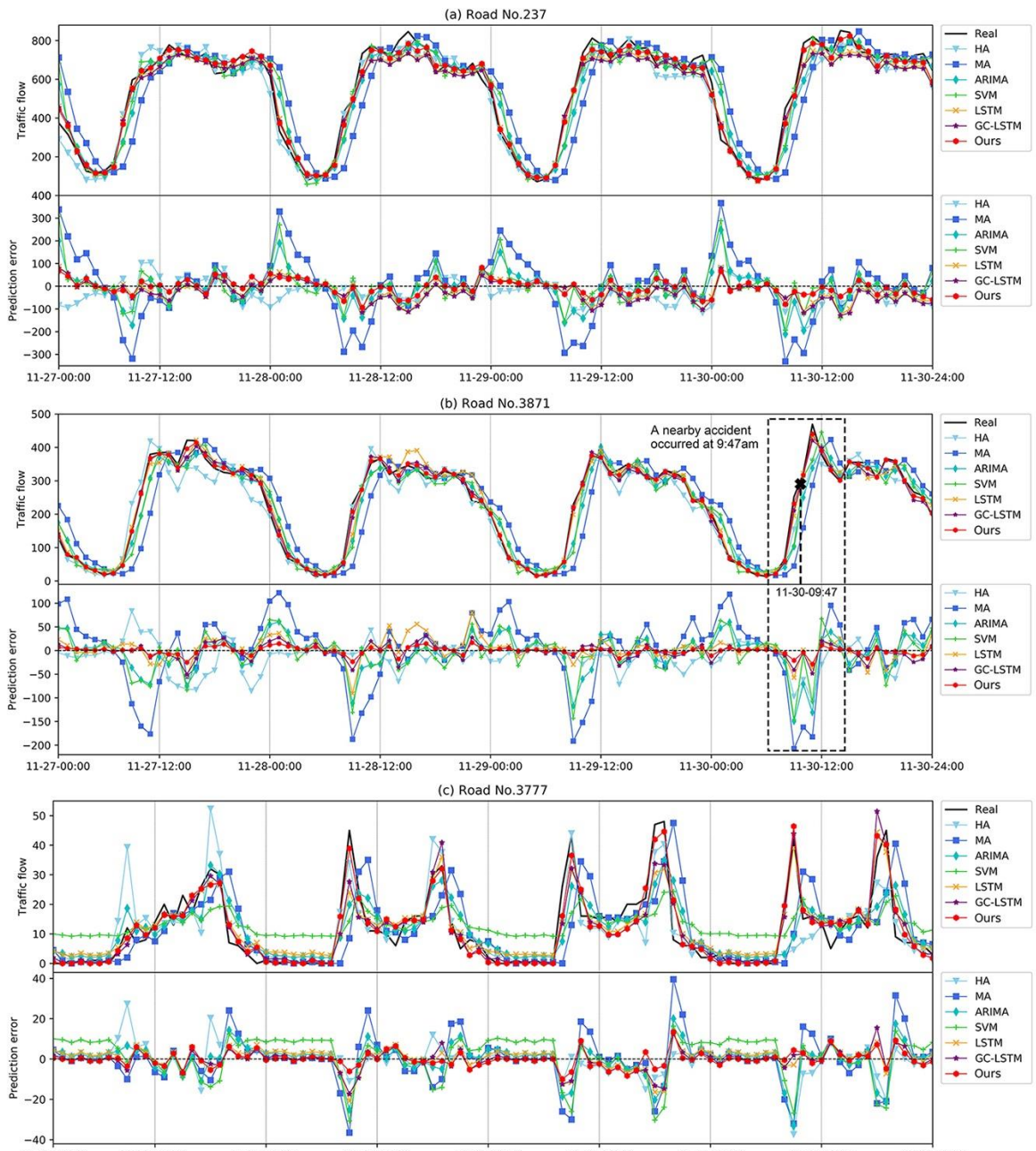

Figure 10 Comparison of the 60-min traffic flow prediction. (a) Road No.237 with heavy traffic capacity. (b) Road No.2246 with medium traffic capacity. (c) Road No.3777 with low traffic capacity.

to the 'zero line'. Compared to the three deep learning models, the three statistical methods and SVR easily generated extremely large prediction errors. Another interesting finding is that the large prediction errors are often found during the traffic flow decreasing from a peak to a valley or increasing from a valley to a peak. In addition, the prediction results of LSTM and GC-LSTM are comparable to STGI-ResNet. However, during peak traffic times, the prediction errors of LSTM-based methods are generally higher than STGI-ResNet. This demonstrates that STGI-ResNet is practical and promising for accurate short-term traffic flow prediction on road networks.

More specifically, according to Figure 10 (b), the three deep learning models perform better than the other models on non-recurrent traffic peak prediction, although no accident indicators are incorporated into the model. This can be because the flow data implicitly reveal the situation in the short term and the deep architectures better capture the dynamics than others. According to Figure 10 (c), SVR performs the worst because we did not train 4098 SVRs for 4098 road segments, but one was trained for all. If each road had its own SVR, the prediction performance might be improved, but the training time would be extremely long. In addition, both the weekday and weekend traffic pattern can be modelled by STGI-ResNet although no weekday or weekend indicators are used in the model.

4.5 Effectiveness and sensitivity analysis

First, STGI-ResNet is compared with its five variants to validate the effectiveness of the proposed framework. The notation and description of the variants are given as follows:

V1: Undirected graph inception network. The adjacent matrix is used to represent the road network as an undirected graph.

V2-V4: The three spatial-temporal graph convolution networks have filters of filter size K=1, 2, and 3, respectively. The models do not have parallel filters with different filter sizes (the inception structure). The models are utilised to justify the necessity of the inception structure.

V5: Graph inception network without the residual short path. It is used to demonstrate the effectiveness of the residual architecture.

For simplicity, the NRMSE is taken as the performance metric, as it removes the scale effect of different flow levels. The comparison results are illustrated in Figure 11. Referring to V1, it shows that representing the road network as a directed graph improves the prediction performance. Observing V2-V4, it is found that without an inception structure, the optimal filter size is K=2 in 10-min and 30-min traffic flow prediction, and K=3 in 60-min traffic flow prediction. This phenomenon shows that the spatial span of the spatial dependency may vary in different prediction time intervals. The NRMSE values of the three variants are all higher than STGI-ResNet's, even if the parameters of V2-V4 are retuned and the parameters of STGI-ResNet are unchanged in different traffic prediction tasks. It validates that the proposed STGI-ResNet is robust and the inception structure can save parameter tuning efforts to some degree, which justifies the necessity of the inception structure. As for V5, its prediction performance is comparable to STGI-ResNet in terms of NRMSE. However, the convergence of STGI-ResNet is much faster than V5 (Figure 12), which proves the residual structure can speed up the training process.

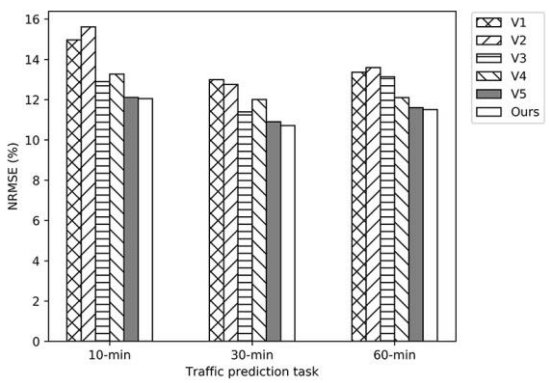

Figure 11 Performance comparison between STGI-ResNet and its five variants in terms of NRMSE

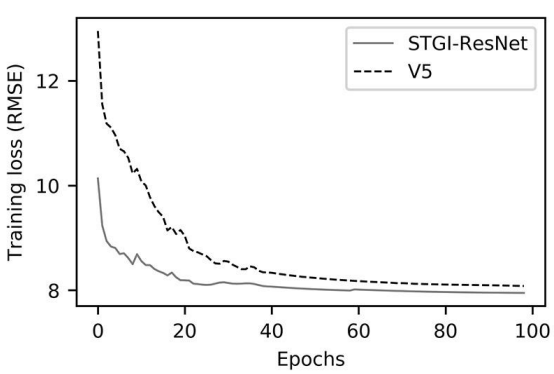

Figure 12 Training process comparison between STGI-ResNet and its variant V5 in 60-min traffic flow prediction.

Second, a sensitivity analysis is conducted to justify the significance of the adaptive fusion of diverse temporal features. Four different combinations of the three temporal features are fed into STGI-ResNet to compare prediction performance, as displayed in Figure 13. $\boldsymbol{X}_t^s$, $\boldsymbol{X}_t^m$, and $\boldsymbol{X}_t^l$ denote the short, medium and long-term temporal features, respectively. Results show that the best performance is achieved when integrating the three types of temporal features. It is suggested that the periodicity of traffic flow is a contributing factor for traffic forecasting. In addition, the NRMSE of '$\boldsymbol{X}_t^s \oplus \boldsymbol{X}_t^l$' is even lower than that of

‘ $\boldsymbol{X}_t^s \oplus \boldsymbol{X}_t^m$ ’, which indicates that the weekly temporal features play a more important role than the daily temporal features.

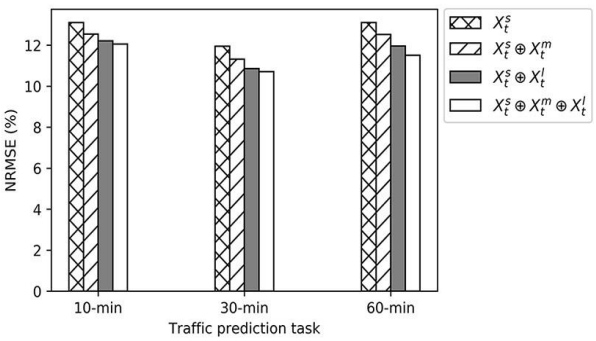

Figure 13 Performance comparison in terms of NRMSE when using different combinations of the three temporal features as the input of STGI-ResNet.

4.6 Prediction error analysis

Figure 14 displays the temporal distribution of the RMSE and NRMSE values of 10-min traffic flow predictions across the four test days. Overall, the prediction quality is stable across the day but changes over time. The absolute prediction errors, i.e., the RMSE values, are small in the very early morning (1:00-7:00) but large during non-sleeping hours (7:00-24:00). The relative errors, i.e., NRMSE values, are the opposite. This is because when the ground truth is small, a small difference between the ground truth and the predicted traffic flow causes a large relative error but a small absolute error. In addition, the largest RMSE and NRMSE appear on Monday. This may be because the daily features used for traffic prediction on Monday are from the previous Sunday, which may have

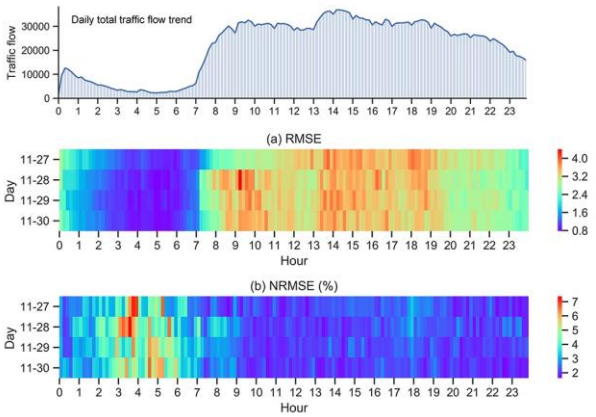

Figure 14 Heatmap of RMSE and NRMSE values

different traffic patterns. However, the mean value of the RMSEs or NRMSEs of the four days has no significant difference, indicating that the STGI-ResNet can model the traffic pattern on different days even without ‘day-of-week’ variables.

Figure 15 (a)-(c) illustrates the spatial distribution of the average traffic flow, RMSE and NRMSE by roads in the 10-min traffic flow prediction task. Generally, the RMSE is positively correlated to the ground truth while the NRMSE is negatively correlated. The reason here is the same as the reason for the temporal error distribution. According to Figure 15 (d)-(f), roads with RMSE values less than 5 account for 90.51% of all roads and those with NRMSE values less than 15% account for 84.94% of all roads.

5 SUMMARY AND CONCLUSIONS

Short-term traffic flow forecasting on a city-wide road

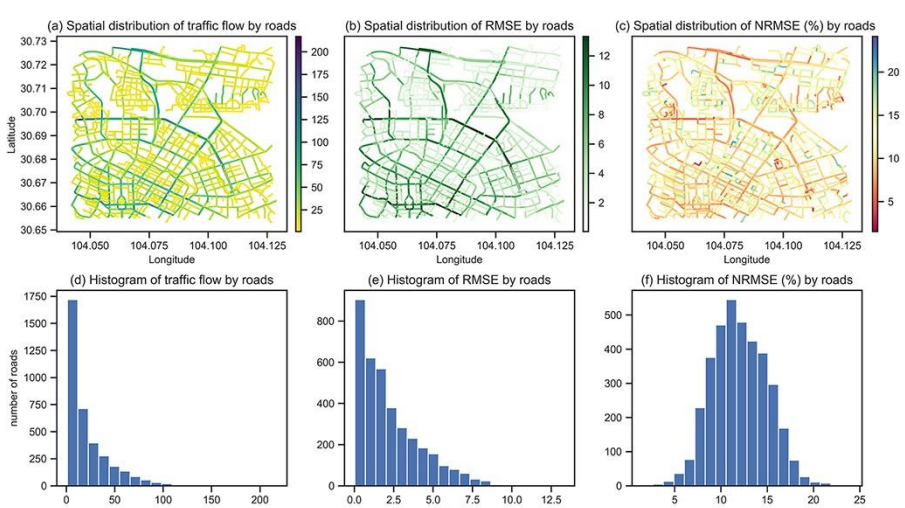

Figure 15 Spatial distribution (a-c) and histograms (d-f) of average traffic flow, RMSE and NRMSE values by roads in 10-min traffic flow prediction task.

network is important for traffic management and control applications. It is challenging due to the spatial-temporal dependencies, the complex network topology and the high computational cost. To overcome the issues, this paper proposes a novel spatial-temporal deep learning framework for large-scale network-based traffic flow prediction. The contributions are summarised below.

This paper represents a road network as a directed 'well-behaved' graph whose nodes are road segments and edges indicate the adjacent relationships. This well-behaved graph enables the road network's topology to be incorporated for traffic forecasting. The dynamics of the network traffic flow are then modelled as a Markov chain on the graph with edge weights determined by the Markov TPM.

Next, a STGI-ResNet model is developed for traffic forecasting. It integrates a novel STGC operator, the residual learning and the inception structure. In STGI-ResNet, the STGC operators adaptively extract temporal features from multiple periodicities and fully utilise spatial information by incorporating the influences of both the upstream and downstream links in the weighted directed graph. The STGC operator is developed based upon the Laplacian function proposed by Chung (2005). It is first used for the convolution operation on graphs, which is only possible once the network has been represented as a weighted well-behaved graph. Additionally, this is the first time the inception residual learning technique has been used for network-structured data and traffic flow prediction.

The approach was evaluated on a large traffic network consisting of 4089 segments in Chengdu, China, for 10-min, 30-min, and 60-min car-hailing traffic flow prediction. Results show that STGI-ResNet significantly improves the prediction accuracy in terms of RMSE, MAE, NRMSE, and MAPE@10 in comparison with various baselines (i.e., HA, MA, ARIMA, SVR, LSTM and GC-LSTM). Regarding the prediction efficiency, the training time of STGI-ResNet is much shorter than ARIMA and LSTM-based methods and it is comparable to SVM. In addition, neither the working days nor weekends are explicitly differentiated, and the traffic incidences are not specifically marked, the proposed model still hold its advantages. This shows that our model could accommodate both weekday/weekend traffic patterns and recurrent/nonrecurrent situations since the flow data implicitly reveals the situations for the short-term traffic.

The proposed model is also compared with its various variants and results indicate that representing the road network as a directed graph improves the prediction performance; the inception structure can greatly improve the robustness of the model as well as saving the parameter tuning efforts; and the residual learning structure enables a quick convergence of model training. Results from sensitivity analysis validate that fusing multiple temporal features enhances the prediction accuracy, implying the periodicities of traffic flow are important contributing factors for traffic flow forecasting. Another interesting finding is that the weekly-periodicity plays a more important role than the daily-periodicity for short-term traffic flow prediction. Finally, by analysing the spatial and temporal distribution of the prediction errors, it shows the proposed model performs well over space during peak and off-peak hours. Overall, despite a slight performance decrease on Monday's traffic flow prediction, the proposed model achieves excellent short-term traffic flow forecasting tasks for different time intervals.

However, this study has some limitations, which will be the directions of future research. First, the traffic data used in this study is the car-hailing traffic flow covering a single month. The model should be tested on other complete traffic flow data, and as more data become available, it will be interesting to explore the effect of other temporal dependencies (e.g. seasonal or yearly periodicity) on prediction accuracy. Second, the graph representation is based on the physical street network topology. An alternative way is to build a virtual graph structure based on the segment flow similarity measured by visibility graph similarity (Ahmadlou and Adeli, 2012; Ahmadlou et al., 2012). It would be interesting to discuss the influence of the graph structure on the traffic prediction. Third, an explicit comparison between the data-driven approaches and the simulation approaches, e.g. DTA models, would be an interesting topic to explore in the future. Finally, the model requires flow data that are available in every time slot and on every road segment for prediction. In the future, models should be developed to predict network traffic flow with missing data.

ACKNOWLEDGMENTS

The authors would like to thank the Editor and the eight anonymous reviewers for their constructive comments to improve the quality of the article. They are also grateful to Dr. James Haworth and Dr. Kun Xie for many valuable discussions about the paper and related work. Didi Chuxing is acknowledged for sharing the datasets used in the paper (https://outreach.didichuxing.com). This work is part of the Consumer Data Research Centre (CDRC) project supported by the UK Economic and Social Research Council (ES/L011840/1). The first author's PhD research is jointly funded by China Scholarship Council (Grant No. 201603170309) and the Dean's Prize from the University College London. The third author's joint PhD research is funded by the China Scholarship Council (Grant No. 201706330020). The authors declare no conflict of interest.

REFERENCES

Abadi, A., Rajabioun, T. & Ioannou, P. A. (2015), Traffic Flow Prediction for Road Transportation Networks

with Limited Traffic Data, *IEEE Transactions on Intelligent Transportation Systems*, **16**(2), 653-662.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. & Isard, M. (2016), Tensorflow: A System for Large-Scale Machine Learning, *OSDI*, pp. 265-283.

Adeli, H. & Jiang, X. (2008), *Intelligent Infrastructure: Neural Networks, Wavelets, and Chaos Theory for Intelligent Transportation Systems and Smart Structures*, Crc Press.

Ahmadlou, M. & Adeli, H. (2012), Visibility Graph Similarity: A New Measure of Generalized Synchronization in Coupled Dynamic Systems, *Physica D: Nonlinear Phenomena*, **241**(4), 326-332.

Ahmadlou, M., Adeli, H. & Adeli, A. (2012), Improved Visibility Graph Fractality with Application for the Diagnosis of Autism Spectrum Disorder, *Physica A: Statistical Mechanics and its Applications*, **391**(20), 4720-4726.

Anwar, T., Liu, C., Vu, H. L. & Islam, M. S. (2015), Roadrank: Traffic Diffusion and Influence Estimation in Dynamic Urban Road Networks, *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ACM, pp. 1671-1674.

Boeing, G. (2017), Osmnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks, *Computers, Environment and Urban Systems*, **65**, 126-139.

Boto-Giralda, D., Díaz-Pernas, F. J., González-Ortega, D., Díez-Higuera, J. F., Antón-Rodríguez, M., Martínez-Zarzuela, M. & Torre-Díez, I. (2010), Wavelet-Based Denoising for Traffic Volume Time Series Forecasting with Self-Organizing Neural Networks, *Computer-Aided Civil and Infrastructure Engineering*, **25**(7), 530-545.

Box, G. E. & Pierce, D. A. (1970), Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models, *Journal of the American Statistical Association*, **65**(332), 1509-1526.

Brualdi, R. A. & Ryser, H. J. (1991), *Combinatorial Matrix Theory*, Springer.

Bruna, J., Zaremba, W., Szlam, A. & LeCun, Y. (2013), Spectral Networks and Locally Connected Networks on Graphs, *arXiv preprint arXiv:1312.6203*.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C. & Sun, J. (2016), A Spatiotemporal Correlative K-Nearest Neighbor Model for Short-Term Traffic Multistep Forecasting, *Transportation Research Part C: Emerging Technologies*, **62**, 21-34.

Cheng, T., Haworth, J. & Wang, J. (2012), Spatio-Temporal Autocorrelation of Road Network Data, *Journal of Geographical Systems*, **14**(4), 389-413.

Cheng, T., Wang, J., Haworth, J., Heydecker, B. & Chow, A. (2014), A Dynamic Spatial Weight Matrix and Localized Space–Time Autoregressive Integrated Moving Average for Network Modeling, *Geographical Analysis*, **46**(1), 75-97.

Chiu, Y.-C., Bottom, J., Mahut, M., Paz, A., Balakrishna, R., Waller, T. & Hicks, J. (2011), Dynamic Traffic Assignment: A Primer, *Transportation Research E-Circular*(E-C153).

Chung, F. (2005), Laplacians and the Cheeger Inequality for Directed Graphs, *Annals of Combinatorics*, **9**(1), 1-19.

Cohen, J. (1988), Statistical Power Analysis for the Behavioral Sciences 2nd Edn, Erlbaum Associates, Hillsdale.

Crisostomi, E., Kirkland, S. & Shorten, R. (2011), A Google-Like Model of Road Network Dynamics and Its Application to Regulation and Control, *International Journal of Control*, **84**(3), 633-651.

Defferrard, M., Bresson, X. & Vandergheynst, P. (2016), Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, *Advances in Neural Information Processing Systems*, pp. 3844-3852.

Dharia, A. & Adeli, H. (2003), Neural Network Model for Rapid Forecasting of Freeway Link Travel Time, *Engineering Applications of Artificial Intelligence*, **16**(7-8), 607-613.

Do, L. N., Taherifar, N. & Vu, H. L. (2018), Survey of Neural Network-Based Models for Short-Term Traffic State Prediction, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1285.

Hagberg, A., Swart, P. & S Chult, D. (2008), Exploring Network Structure, Dynamics, and Function Using Networkx, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hammond, D. K., Vandergheynst, P. & Gribonval, R. (2011), Wavelets on Graphs Via Spectral Graph Theory, *Applied and Computational Harmonic Analysis*, **30**(2), 129-150.

Hashemi, H. & Abdelghany, K. (2015), Real-Time Traffic Network State Prediction for Proactive Traffic Management: Simulation Experiments and Sensitivity Analysis, *Transportation Research Record: Journal of the Transportation Research Board*(2491), 22-31.

Hashemi, H. & Abdelghany, K. (2018), End-to-End Deep Learning Methodology for Real-Time Traffic Network Management, *Computer - Aided Civil and Infrastructure Engineering*.

Haworth, J., Shawe-Taylor, J., Cheng, T. & Wang, J. (2014), Local Online Kernel Ridge Regression for Forecasting of Urban Travel Times, *Transportation Research Part C: Emerging Technologies*, **46**, 151-178.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep Residual Learning for Image Recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778.

Heymann, S. (2014), Gephi, *Encyclopedia of Social Network Analysis and Mining*, Springer, pp. 612-625.

Hofmann, H., Wickham, H. & Kafadar, K. (2017), Value Plots: Boxplots for Large Data, *Journal of Computational and Graphical Statistics*, **26**(3), 469-477.

Jiang, X. & Adeli, H. (2004), Wavelet Packet - Autocorrelation Function Method for Traffic Flow Pattern Analysis, *Computer - Aided Civil and Infrastructure Engineering*, **19**(5), 324-337.

Jiang, X. & Adeli, H. (2005), Dynamic Wavelet Neural Network Model for Traffic Flow Forecasting, *Journal of transportation engineering*, **131**(10), 771-779.

Kingma, D. P. & Ba, J. (2014), Adam: A Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, pp. 1097-1105.

LeCun, Y., Bengio, Y. & Hinton, G. (2015), Deep Learning, *Nature*, **521**(7553), 436.

Lippi, M., Bertini, M. & Frasconi, P. (2013), Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning, *IEEE Transactions on Intelligent Transportation Systems*, **14**(2), 871-882.

Lv, Y., Duan, Y., Kang, W., Li, Z. & Wang, F.-Y. (2015), Traffic Flow Prediction with Big Data: A Deep Learning Approach, *IEEE Transactions on Intelligent Transportation Systems*, **16**(2), 865-873.

Ma, Z., Xing, J., Mesbah, M. & Ferreira, L. (2014), Predicting Short-Term Bus Passenger Demand Using a Pattern Hybrid Approach, *Transportation Research Part C: Emerging Technologies*, **39**, 148-163.

Manley, E., Cheng, T., Penn, A. & Emmonds, A. (2014), A Framework for Simulating Large-Scale Complex Urban Traffic Dynamics through Hybrid Agent-Based Modelling, *Computers, Environment and Urban Systems*, **44**, 27-36.

May, A. D. (1990), *Traffic Flow Fundamentals*.

Min, X., Hu, J., Chen, Q., Zhang, T. & Zhang, Y. (2009), Short-Term Traffic Flow Forecasting of Urban Network Based on Dynamic Starima Model, *Intelligent Transportation Systems, 2009. ITSC'09. 12th International IEEE Conference on*, IEEE, pp. 1-6.

Nabian, M. A. & Meidani, H. (2018), Deep Learning for Accelerated Seismic Reliability Analysis of Transportation Networks, *Computer-Aided Civil and Infrastructure Engineering*, **33**(6), 443-458.

Newson, P. & Krumm, J. (2009), Hidden Markov Map Matching through Noise and Sparseness, *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, ACM, pp. 336-343.

Niepert, M., Ahmed, M. & Kutzkov, K. (2016), Learning Convolutional Neural Networks for Graphs, *International conference on machine learning*, pp. 2014-2023.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. & Dubourg, V. (2011), Scikit-Learn: Machine Learning in Python, *Journal of machine learning research*, **12**(Oct), 2825-2830.

Polson, N. G. & Sokolov, V. O. (2017), Deep Learning for Short-Term Traffic Flow Prediction, *Transportation Research Part C: Emerging Technologies*, **79**, 1-17.

Ren, Y., Cheng, T. & Zhang, Y. (2019), A Deep Spatio-Temporal Residual Neural Network for Road-Network-Based Data Modelling, *International journal of geographical information science*.

Shu, Y., Yu, M., Liu, J. & Yang, O. W. (2003), Wireless Traffic Modeling and Prediction Using Seasonal Arima Models, *Communications, 2003. ICC'03. IEEE International Conference on*, IEEE, pp. 1675-1679.

Smith, B. L. & Oswald, R. K. (2003), Meeting Real–Time Traffic Flow Forecasting Requirements with Imprecise Computations, *Computer - Aided Civil and Infrastructure Engineering*, **18**(3), 201-213.

Stathopoulos, A., Dimitriou, L. & Tsekeris, T. (2008), Fuzzy Modeling Approach for Combined Forecasting of Urban Traffic Flow, *Computer-Aided Civil and Infrastructure Engineering*, **23**(7), 521-535.

Sullivan, G. M. & Feinn, R. (2012), Using Effect Size-or Why the P Value Is Not Enough, *Journal of graduate medical education*, **4**(3), 279-282.

Sun, S., Zhang, C. & Yu, G. (2006), A Bayesian Network Approach to Traffic Flow Forecasting, *IEEE Transactions on Intelligent Transportation Systems*, **7**(1), 124-132.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015), Going Deeper with Convolutions, Cvpr.

Vlahogianni, E. I., Karlaftis, M. G. & Golias, J. C. (2007), Spatio-Temporal Short-Term Urban Traffic Volume Forecasting Using Genetically Optimized Modular Networks, *Computer-Aided Civil and Infrastructure Engineering*, **22**(5), 317-325.

Vlahogianni, E. I., Karlaftis, M. G. & Golias, J. C. (2014), Short-Term Traffic Forecasting: Where We Are and Where We're Going, *Transportation Research Part C: Emerging Technologies*, **43**, 3-19.

Williams, B. (2001), Multivariate Vehicular Traffic Flow Prediction: Evaluation of Arimax Modeling, *Transportation Research Record: Journal of the Transportation Research Board*(1776), 194-200.

Williams, B. M. & Hoel, L. A. (2003), Modeling and Forecasting Vehicular Traffic Flow as a Seasonal

Arima Process: Theoretical Basis and Empirical Results, *Journal of transportation engineering*, **129**(6), 664-672.

Wu, Y. & Tan, H. (2016), Short-Term Traffic Flow Forecasting with Spatial-Temporal Correlation in a Hybrid Deep Learning Framework, *arXiv preprint arXiv:1612.01022*.

Xie, Y., Zhang, Y. & Ye, Z. (2007), Short-Term Traffic Volume Forecasting Using Kalman Filter with Discrete Wavelet Decomposition, *Computer-Aided Civil and Infrastructure Engineering*, **22**(5), 326-334.

Yu, B., Yin, H. & Zhu, Z. (2017), Spatio-Temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting, *arXiv preprint arXiv:1709.04875*.

Yu, G., Hu, J., Zhang, C., Zhuang, L. & Song, J. (2003), Short-Term Traffic Flow Forecasting Based on Markov Chain Model, *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, IEEE, pp. 208-212.

Yu, H., Wu, Z., Wang, S., Wang, Y. & Ma, X. (2017), Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks, *Sensors*, **17**(7), 1501.

Zhang, J., Zheng, Y. & Qi, D. (2017), Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction, *AAAI*, pp. 1655-1661.

Zhang, J., Zheng, Y., Qi, D., Li, R. & Yi, X. (2016), Dnn-Based Prediction Model for Spatio-Temporal Data, *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, pp. 92.

Zhang, Y. & Cheng, T. (2019), A Deep Learning Approach to Infer Employment Status of Passengers by Using Smart Card Data, *IEEE Transactions on Intelligent Transportation Systems*.

Zhao, Z., Chen, W., Wu, X., Chen, P. C. & Liu, J. (2017), Lstm Network: A Deep Learning Approach for Short-Term Traffic Forecast, *IET Intelligent Transport Systems*, **11**(2), 68-75.

Zou, H., Yue, Y., Li, Q. & Shi, Y. (2010), A Spatial Analysis Approach for Describing Spatial Pattern of Urban Traffic State, *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, IEEE, pp. 557-562.