

Article

A Temporal Directed Graph Convolution Network for Traffic Forecasting Using Taxi Trajectory Data

Kaiqi Chen, Min Deng and Yan Shi *

Department of Geo-Informatics, Central South University, Changsha 410083, China; chenkaiki@csu.edu.cn (K.C.); dengmin@csu.edu.cn (M.D.)

* Correspondence: csu_shiy@csu.edu.cn

Abstract: Traffic forecasting plays a vital role in intelligent transportation systems and is of great significance for traffic management. The main issue of traffic forecasting is how to model spatial and temporal dependence. Current state-of-the-art methods tend to apply deep learning models; these methods are unexplainable and ignore the a priori characteristics of traffic flow. To address these issues, a temporal directed graph convolution network (T-DGCN) is proposed. A directed graph is first constructed to model the movement characteristics of vehicles, and based on this, a directed graph convolution operator is used to capture spatial dependence. For temporal dependence, we couple a keyframe sequence and transformer to learn the tendencies and periodicities of traffic flow. Using a real-world dataset, we confirm the superior performance of the T-DGCN through comparative experiments. Moreover, a detailed discussion is presented to provide the path of reasoning from the data to the model design to the conclusions.

Keywords: traffic flow forecasting; Markov chain; directed graph convolution; transformer structure; spatial dependence; temporal dependence



Citation: Chen, K.; Deng, M.; Shi, Y. A Temporal Directed Graph Convolution Network for Traffic Forecasting Using Taxi Trajectory Data. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 624. <https://doi.org/10.3390/ijgi10090624>

Academic Editor: Wolfgang Kainz

Received: 16 July 2021

Accepted: 13 September 2021

Published: 17 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Traffic flow forecasting aims to estimate traffic conditions (e.g., the velocities or travel time of traffic flow) of each segment on road networks in future time periods based on historical information [1]. It has played an important role in intelligent transportation systems (ITSs) on account of its extensive applications in urban transportation [2]. For instance, Google Maps can provide users with high-quality route planning and navigation services with the aid of traffic forecasting for the purpose of avoiding traffic congestion [3]. Despite the massive efforts made by relevant studies, high-precision and high-reliability traffic forecasting is still subject to the nonlinear dependence of traffic flow variables in the dimensions of both space and time [1,2,4–6].

On the one hand, the time series of traffic flow variables generally present significant temporal dependence in both the short and long term [4]. Specifically, traffic conditions are highly correlated with those observed at adjacent times, and the short-term correlations are gradually delayed with increasing temporal distance. Additionally, the periodicity of traffic flow series on multiple temporal scales can be modeled as long-term temporal dependence. On the other hand, relevant studies have confirmed the existence of dependence between the traffic flow variables observed on topologically connected road segments with certain time lags; this is defined as spatiotemporal dependence [1,2,4]. In traffic applications such as autonomous driving and signal light control, model-based traffic simulators (e.g., LWR and PW) have been widely employed to simulate various traffic flows on road networks by considering spatiotemporal dependences [7]. However, in spite of their effectiveness in modeling the evolution of traffic flow on road networks, the lack of vehicle behavior information combined with the high costs of computational time fundamentally limit the applications of model-based traffic simulators in real-time traffic forecasting on large-scale urban road networks [8]. Nowadays, the increasing availability of discrete trace points

recoded by vehicle-mounted GPS enables the characterization of time-varying traffic flow states at the road segment level [9]. In this context, large amounts of data-driven models have been specifically designed for the task of traffic flow forecasting [1,2,4–6]. Currently, there are two alternative strategies for handling spatiotemporal dependence in traffic flow forecasting based on data-driven models. The first is constructing machine learning models by modeling spatiotemporal dependence as parameters to be estimated, such as the space–time auto-regressive integrated moving average (ST-ARIMA) model [10]. To extract implicit features derived from spatiotemporal dependence, a set of deep learning-based forecasting methods have been designed by coupling a convolutional neural network (CNN) with a recurrent neural network (RNN), such as CNN-Long Short-Term Memory (CNN-LSTM) models [11]. However, the requirement of grid partitioning in Euclidean space limits the capacity of traditional CNNs to accurately capture the spatial dependence among road segments to a larger extent. For cases such as this, recent studies have constructed an undirected graph structure to express the topological relationships between road segments, and it was based on this that a graph convolution neural network was employed to implement traffic flow forecasting [1,2,4–6,12–14].

According to related studies in the field of transportation, there are a total of three elements, i.e., drivers, vehicles, and road segments, that constitute a transportation system [15]. This means that the traffic flow on a road network is determined by both the moving characteristics of the vehicles and the driving rules on the road segments. In the road network shown in Figure 1, the flow direction and volume of moving vehicles are represented by arrows and dotted lines, respectively. Segments 4 and 2 are both spatially adjacent to segment 1, so segment pairs 4–1 and 2–1 have a consistent topological structure. However, the two segment pairs do not necessarily share similar traffic flow distributions due to the diverse driving directions of the vehicles. In addition, the driving rules on road segments cannot be represented by the topology. For instance, segments 1, 4, and 7 are all one-way roads with only one allowable driving direction, while vehicles are only allowed to turn around on segment 3 despite it being topologically connected with segment 1. There is a similar case in which the vehicles on segment 4 are prohibited from turning left into the adjacent segment 6. Based on the above discussions, we can determine that the diversity of driving directions and rules on road segments poses great challenges to current methods of anisotropic spatial dependence modeling and reliable traffic condition forecasting.

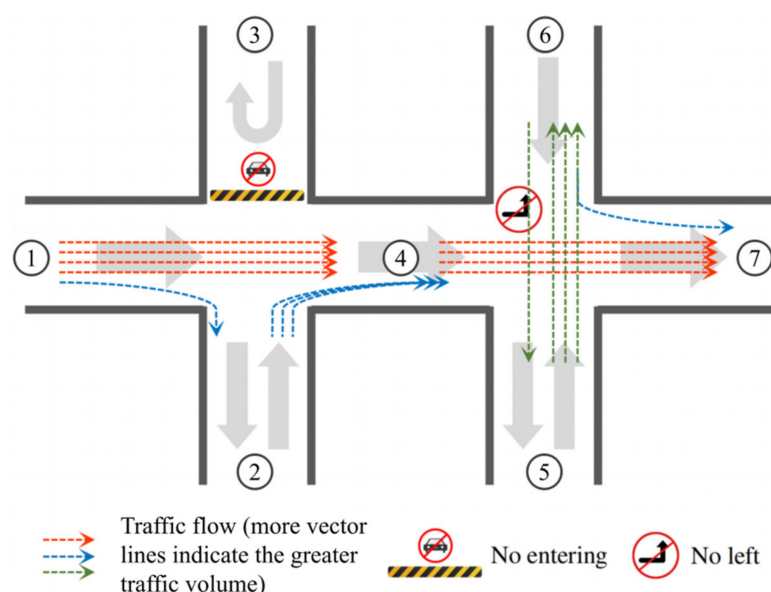


Figure 1. An example of the diversity of traffic patterns on road segments.

To overcome the aforementioned challenges, this study develops a new traffic flow forecasting method by constructing a temporal directed graph convolution network (T-DGCN)

with the combined consideration of multiterm temporal dependence and vehicle movement patterns on road networks. The main contributions of this study include the following three aspects:

- (1) A directed graph is constructed based on the Markov transition probabilities of traffic flow to model the spatial dependence in an objective way, while a new spectral directed graph convolution operator is designed to address the asymmetry of the directed graph.
- (2) A transformer architecture with a novel global position encoding strategy is integrated to capture multiterm temporal dependence, with the aim of improving the interpretability and validity of the forecasting model.
- (3) Comparative experiments on real-world datasets are conducted to provide convincing evidence for the superior performance of the proposed method in traffic flow forecasting.

The remainder of this article is organized as follows: Section 2 gives a full review of the relevant research. Section 3 defines the critical problem and presents the proposed T-DGCN. In Section 4, comparative experiments on real-world datasets are performed to validate the superiority of the proposed method, while Section 5 provides an attribution analysis of the experimental results. Finally, we conclude this study and provide future research directions in Section 6.

2. Related Work

With the extensive utilization of data mining models in traffic flow analysis during the past few decades, an enormous number of methods have been specifically designed for traffic flow forecasting based on machine learning models or deep neural networks [1,2,5,6]. These two types of methods are reviewed in detail in the following.

Machine learning-based spatiotemporal forecasting models aim to estimate the target spatial variable values at future times through parameter training with the constraint of artificially defined spatiotemporal dependence.

With the successful use of the autoregressive integrated moving average model (ARIMA) in time series forecasting [16], Hamed et al. [17] initially introduced this machine learning model to urban traffic volume forecasting. On this basis, extensively modified ARIMA models were successively proposed to improve traffic flow forecasting accuracy. For instance, the Kohonen ARIMA model used a Kohonen self-organizing map to separate the initial time series into homogeneous fragments to track the long-term temporal dependence [18]. Guo et al. [19] integrated the Kalman filter with the generalized auto-regressive conditional heteroskedasticity model to improve the performance of short-term traffic flow forecasting. In addition to ARIMA-based models, support vector regression (SVR)-based models also have outstanding performance in traffic flow forecasting [20]. For instance, Su et al. [21] utilized the incremental support vector regression (ISVR) model to implement the real-time forecasting of traffic flow states, and Gopi et al. [22] proposed a Bayesian support vector regression model, which can provide error bars along with predicted traffic states. Besides this, other common machine learning models have also been applied to the task of traffic flow forecasting. Yin et al. [23] combined fuzzy clustering with a neural network to design a fuzzy neural traffic flow forecasting approach. Cai et al. [24] constructed an improved K-nearest neighbor (KNN) graph to optimize short-term traffic flow forecasting results with the help of spatiotemporal correlation modeling. Sun et al. [25] proposed a Bayesian network-based approach to maximize the joint probability distribution between the historical traffic flow states used as antecedents and the future states to be estimated.

Considering the subjectivity in the measurement of spatiotemporal proximity effects, existing machine learning-based models are greatly limited in capturing the underlying dependence in multiple ranges in space and time. Compared to traditional machine learning models, deep neural networks have self-learning capacity without the input of any artificially extracted features. This powerful learning capability has enabled various

types of deep neural networks to be utilized in the forecasting of traffic flow on road networks [1,3,6].

In essence, the traffic flow on road networks can be classified as a kind of space–time sequence data [2]. Specifically, for the traffic flow sequence on any road segment, the RNN and its variants, such as the long short-term memory (LSTM) unit [26] and the gated recurrent unit (GRU) [27], were widely utilized to learn the dependence between time-varying traffic flow states. For example, Ma et al. [28] developed a forecasting approach to analyze the evolution of traffic congestion by coupling deep restricted Boltzmann machines with an RNN that inherits congestion prediction abilities. Tian et al. [29] utilized a LSTM to determine the optimal time lags dynamically and to achieve higher forecasting accuracy and better generalization. Focusing on the spatial dimension, Wu and Tan [30] mapped the recorded traffic flow states into regular grids divided from the study area to stack sequential images in chronological order. This can leverage the local receptive field in a CNN to capture the spatial dependence of traffic flow states in planar space. However, it is well known that the transfer of traffic flow is rigidly constrained on road networks in reality, so it is necessary to measure the spatiotemporal dependence of traffic flow in the road network space. To address this issue, most studies have used each segment or sensor as the minimum spatial unit and have organized the road network into a graph based on the topological relationships between segments [1,3,6]. In this way, the idea of graph convolution can be employed to extract spatially dependent embedded features from the graph structure. For example, Zhao et al. [2] designed a T-GCN model that introduced 1st ChebNet [12] to model the spatial dependence of traffic networks. Li et al. [13] proposed a diffusion convolutional recurrent neural network (DCRNN) model that performed a diffusion graph convolution on a traffic network to aggregate the spatial neighborhood information of each node and captured long-term temporal dependence using a RNN. Yu et al. [31] constructed a 3D graph convolution network that could simultaneously capture spatial and temporal dependence in the process of feature learning.

As mentioned in Section 1, although existing methods have utilized the topological structure of traffic networks to model spatial dependence, it is still necessary to quantitatively represent the movement patterns and driving rules of vehicles on road networks to improve the rationality of traffic flow forecasting. In terms of temporal dependence, in the majority of current RNN-based strategies, the specific modeling of the tendency and periodicity characteristics in the time-varying process of traffic flow states is insufficient. That is, a large number of relevant historical observations have not yet been sufficiently exploited in an appropriate way, which restricts the accuracy of traffic flow forecasting. To solve these two problems, this study designs a new method by coupling a directed graph convolution network with a transformer structure to model anisotropic spatial dependence and multiterm temporal dependence for the purposes of self-learning the underlying spatiotemporal features of traffic flow states to obtain high-precision forecasting results.

3. Method

This section describes the proposed new traffic flow forecasting method. Specifically, a directed traffic graph is first constructed by using a Markov chain-based strategy, as described in Section 3.1; based on that, a spectral directed graph convolution kernel is used to capture anisotropic spatial dependence, as presented in Section 3.2. In Section 3.3, we design a keyframe sequence and employ a transformer structure for the extraction of multiterm temporal dependence features. Finally, in Section 3.4, we build the T-DGCN by assembling the spatial and temporal dependency learning modules.

3.1. A Markov Chain-Based Strategy for Constructing a Directed Traffic Graph

In this study, considering the directivity of traffic flow, we specifically represent the traffic information on road networks using a graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E}, P)$, where the road segments and intersections constitute the node set $\mathcal{V} = \{v^1, v^2, v^M\}$ and the edge set $\mathcal{E} = \{e^1, e^2, \dots, e^N\}$, respectively. In this way, the traffic flow states on the road

network can be abstracted into a tensor $\mathcal{X} \in \mathbb{R}^{M \times T \times C}$, where M , T , and C denote the number of segments, timestamps, and traffic flow feature dimensions, respectively. For the edges in $\mathcal{G}$, the majority of the current related studies generally quantify the topological relationships between the road segments to obtain a symmetrical adjacency matrix P . To further reflect the anisotropy in traffic flow spatial dependence, this study constructs a Markov chain-based directed graph to describe the transition probabilities of the traffic flow at intersections.

From the perspective of a discrete stochastic process, the transition of the traffic flow between any pair of nodes in $\mathcal{G}$ can be considered to follow the hypothesis of a random walk [32]. Let rs_t denote the located road segment of traffic flow at timestamp t . The transition process can be modeled using a Markov chain, i.e.,

$$P[(rs_0 \rightarrow rs_1 \rightarrow \dots \rightarrow rs_{t-1} \rightarrow rs_t) \rightarrow rs_{t+1}] = P[rs_t \rightarrow rs_{t+1}] \quad (1)$$

This means that the current traffic flow states can entirely determine the future distribution of traffic flow on road networks. On this basis, given any two nodes v_i and v_j , we can calculate the transition probability of traffic flow from v_i to v_j as $p_{ij} = P[(rs_t = v_i) \rightarrow (rs_{t+1} = v_j)]$ and can construct the following Markov transition matrix:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MM} \end{pmatrix} \in \mathbb{R}^{M \times M} \quad (2)$$

We recombine the road nodes into a graph structure according to the transition matrix P . To obtain the transition matrix, we define an intermediate variable γ_{ij} to denote the number of vehicles that move from segment v_i to v_j and form the following matrix Γ :

$$\Gamma = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1M} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{M1} & \gamma_{M2} & \cdots & \gamma_{MM} \end{pmatrix} \in \mathbb{R}^{M \times M} \quad (3)$$

On this basis, the transition matrix P can be expressed as

$$P = \text{diag}(\Gamma \mathbf{1})^{-1} \Gamma \quad (4)$$

Here, $\mathbf{1}$ is a vector of all ones. In this transition matrix, each element $p_{ij} = \frac{\gamma_{ij}}{\sum_{k=1}^M \gamma_{ik}}$ essentially quantifies the moving probability of traffic flow from v_i to v_j .

3.2. A Directed Graph Convolution Kernel for Capturing Spatial Dependence

Regarding the forecasting of space–time sequences organized using graph structures, e.g., traffic flow series, the spectral graph convolution neural network has shown powerful performance in learning dependence features on multiple spatial scales [12]. However, most spectral-based methods are limited to only working on undirected graphs [33]. According to spectral graph theory, it is necessary to find a directed Laplacian operator to implement the convolution operation on a constructed directed traffic graph without the loss of direction information. In this case, we leverage the Perron–Frobenius theorem to embed a directed Laplacian operator into the graph convolution neural network [34].

Let $r_{ij}(n) = P[(v_i \rightarrow \dots \rightarrow v_j)_n]$ denote the probability that the state changes from v_i to v_j after step n ; this term can be calculated using the following Chapman–Kolmogorov Equations [34]:

$$r_{ij}(n) = \sum_{k=1}^M r_{ik}(n-1)p_{kj} \quad (5)$$

The connectivity of the urban road network indicates that any two road segments can be connected through the flow of vehicles ($\forall v_i, v_j \in \mathcal{V}, \exists n$ that $r_{ij}(n) > 0$), which means that the Markov chain-based directed graph has the characteristic of strong connections. According to the steady-state convergence theorem, the stationary distribution of traffic flow states on road networks can be denoted as [34]:

$$\pi = \pi_0 P^n \quad (6)$$

Here, π_0 denotes the initial vector of traffic flow states, while n tends to positive infinity. We can treat π as a Perron vector according to the Perron–Frobenius theorem to define the Laplacian operator of a directed graph, i.e.,

$$\mathcal{L} = I - \Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}}, \text{ where } \Pi = \text{diag}(\pi) \quad (7)$$

For the asymmetric matrix, the corresponding symmetric Laplacian can be expressed as [33]

$$\mathcal{L}^{sym} = I - \frac{1}{2} \left(\Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} + \Pi^{-\frac{1}{2}} P^T \Pi^{\frac{1}{2}} \right) \quad (8)$$

In this way, we symmetrize the original directed traffic graph, so we can obtain the graph convolution kernel $x * g_\theta = U(U^T x \odot U^T g_\theta)$. Then, this filter can be approximated using Chebyshev polynomials [33]

$$x * g_\theta \approx \sum_{k=0}^{K-1} \theta_k T(\tilde{\mathcal{L}}^{sym}) x \quad (9)$$

where $\tilde{\mathcal{L}}^{sym} = \frac{2}{\lambda_{max}^{sym}} \mathcal{L}^{sym} - I$ is the rescaled form of $\mathcal{L}^{sym}$ for locating eigenvalues within $[-1, 1]$. Let $K = 2$ and $\theta = \theta_0 = -\theta_1$ and further approximate the largest eigenvalue of $\mathcal{L}^{sym}$ as $\lambda_{max}^{sym} \approx 2$ according to [12]. The filter can be simplified as

$$x * g_\theta \approx \theta \left[I + \frac{1}{2} \left(\Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} + \Pi^{-\frac{1}{2}} P^T \Pi^{\frac{1}{2}} \right) \right] x \quad (10)$$

To alleviate the problems of exploding and vanishing gradients, Kipf and Welling [12] used a renormalization strategy, i.e., $I + D^{-1/2} A D^{-1/2} \rightarrow \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$, by adding a self-loop to each node $\tilde{A} = A + I$. Due to the self-loop structure of the Markov chain-based directed graph, we utilize another renormalization strategy. Let $\theta = \frac{2}{2 - \lambda_{max}^{sym}} \theta_0 = -\frac{2}{\lambda_{max}^{sym}} \theta_1$, and Equation (10) can be redefined as

$$x * g_\theta \approx \theta \left[\frac{1}{2} \left(\Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} + \Pi^{-\frac{1}{2}} P^T \Pi^{\frac{1}{2}} \right) \right] \quad (11)$$

Finally, the directed graph convolution layer can be represented as

$$Z = \frac{1}{2} \left(\Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} + \Pi^{-\frac{1}{2}} P^T \Pi^{\frac{1}{2}} \right) X \Theta \quad (12)$$

Here, $\theta \in \mathbb{R}^{d_{in} \times d_{model}}$ is the learnable parameter, and d_{in} and d_{model} denote the dimensions of the input features and hidden features, respectively.

3.3. A Transformer Structure for Learning Temporal Dependence Features

In addition to the dependence of traffic flow in the space dimension, other critical issues exist that need to be addressed in traffic flow forecasting, that is, extracting dependence features between traffic flow states at distinct timestamps [2]. Faced with this problem, the most widely used solution at present is the RNN [1]. However, current RNN-based models were not specifically designed considering the inherent time-variant characteristics of traffic flow states and tend to be overly complex, including a large number

of learnable parameters. On the basis of prior knowledge, we design keyframe sequences to organize the original data and leverage a transformer structure to extract multiterm temporal dependency features.

As discussed in Section 1, the temporal dependence of traffic flow states mainly includes short-term and long-term states; these indicate the tendencies and periodicities of traffic flow time series, respectively. For each road segment at t , we first define the tendency-related sequence as $\mathcal{X}_t(t) = \left\{ \mathcal{X}^{(t-\Delta t)} \mid \Delta t \leq tl, \Delta t \in \mathbb{N}^+ \right\}$ by using a time lag parameter tl . In addition, current relevant work generally regards the periodicity as the correlations between the observations at t and those at the corresponding times in the previous few days or weeks [35]. Considering the slight fluctuation in the variation cycle regarding traffic flow states, this study introduces a time window parameter tw to define an interval around each periodic timestamp, within which the periodicity can be refined by embedding the local tendencies. Then, the periodicity-related sequence can be defined as $\mathcal{X}_p(t) = \left\{ \mathcal{X}^{(t-\Delta t)} \mid \Delta t \in [nT_p - tw, nT_p + tw], n \leq N_p \text{ and } n, N_p, tw \in \mathbb{N}^+ \right\}$ within N_p cycles, where T_p denotes the length of one cycle. $\mathcal{X}_t(t)$ and $\mathcal{X}_p(t)$ form the keyframe sequence $\mathcal{X}_k(t)$ at timestamp t . By inputting each member of $\mathcal{X}_k(t)$ to the directed graph convolution layer in parallel, we can capture a spatial feature sequence tensor $\mathcal{F}(t) \in \mathbb{R}^{M \times (tl + N_p(1+2tw)) \times d_{model}}$. To facilitate the capture of time- and space-varying temporal dependence, we further employ daily periodic position embedding [36] and node2vec embedding [37] strategies to encode the absolute time and space information for each timestamp and each road segment. After that, the tensor $\mathcal{F}(t)$ can be integrated with the space-time information by elementwise addition operations.

Targeting the spatial feature tensor $\mathcal{F}(t)$, we use self-attention to calculate the implicit multirelationships on the keyframe sequence of each road segment at timestamp t . Basically, three subspaces, namely the query subspace $Q^s \in \mathbb{R}^{d_{model} \times d_k}$, the key subspace $K^s \in \mathbb{R}^{d_{model} \times d_k}$, and the value subspace $V^s \in \mathbb{R}^{d_{model} \times d_v}$, are obtained by performing linear mapping operations on $\mathcal{F}(t)$, i.e.,

$$Q^s = \mathcal{F}(t) \times W_q^s, \quad K^s = \mathcal{F}(t) \times W_k^s, \quad V^s = \mathcal{F}(t) \times W_v^s \quad (13)$$

Here, W_q^s , W_k^s , and W_v^s are learnable parameters. To better capture multiterm temporal dependence, multihead attention is further introduced by concatenating N_h single attention heads, i.e.,

$$Head = \left(Head^{(1)} \circ Head^{(2)} \circ \dots \circ Head^{(H)} \right) \quad (14)$$

where $Head^{(h)} = \text{softmax}\left(S^{(h)}\right)V^h$, and $S^{(h)} = \frac{Q^h(K^h)^T}{\sqrt{d_k^h}}$. Note that ' $\circ$ ' denotes a concatenation operator. After that, a new tensor $\mathcal{F}_{out}(t)$ that contains the spatial-temporal features can be produced using a learnable parameter $W^0 \in \mathbb{R}^{(Hd_v) \times (d_{model})}$,

$$\mathcal{F}_{out}(t) = Head \times W \quad (15)$$

On this basis, we can construct the transformer structure by the classical encoder-decoder method [38] to implement traffic flow forecasting. As shown in Figure 2, both the encoder and the decoder contain N_{cell} identical cells. Each identical cell is mainly constituted by a multihead attention layer and a keyframe-wise fully connected feed-forward layer. Residual connections and normalization layers are also integrated. Note that the decoder cell has one more multihead attention layer than the encoder cell, which has the function of calculating the multihead attention over the features of the historical keyframes and the forecasted ones.

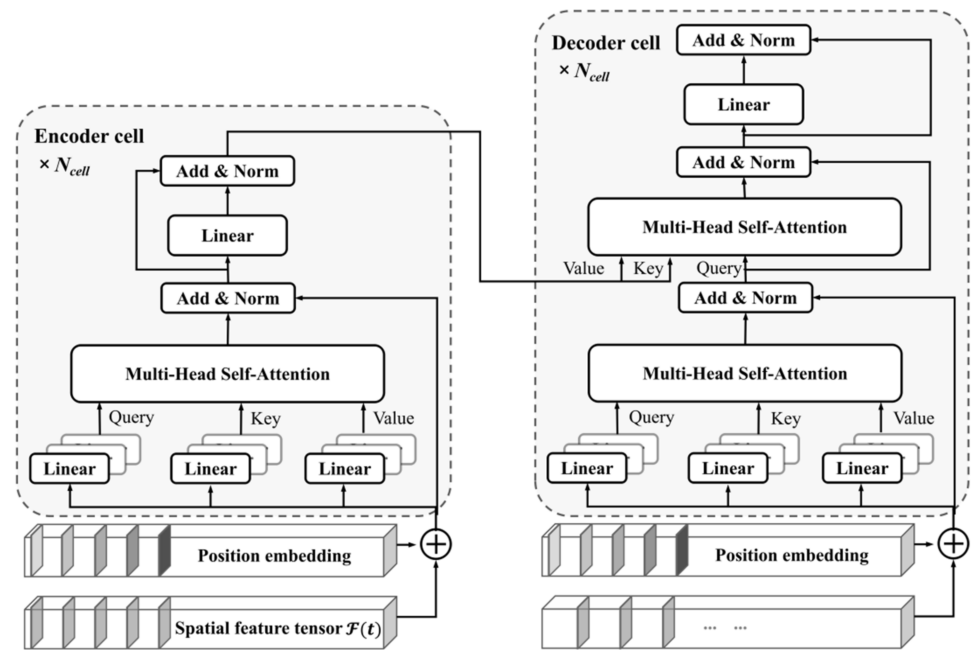


Figure 2. The basic structure of the encoder and decoder. Each encoder (decoder) is composed of N_{cell} encoder cells (decoder cells).

3.4. Temporal Directed Graph Convolution Network (T-DGCN)

With the integration of the Markov chain-based directed graph convolution layer with the transformer structure-based encoder-decoder layer, Figure 3 gives the overall architecture of the proposed T-DGCN. Specifically, for the keyframe sequences of each road segment, two Markov-based directed graph convolution layers are used to capture keyframe-wise spatial dependence to construct the spatial feature tensor $\mathcal{F}(t)$. The network further utilizes the transformer structure-based encoder-decoder layer to learn multiterm temporal dependence features from $\mathcal{F}(t)$. The forecasted results are ultimately output from a fully connected layer.

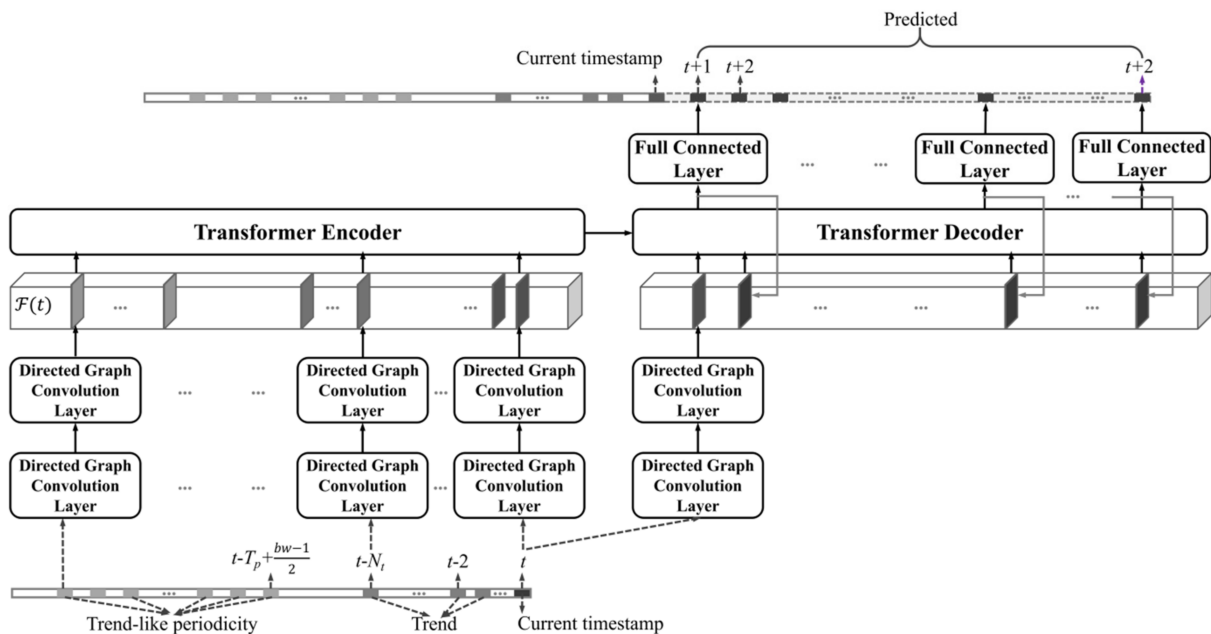


Figure 3. The overall architecture of the T-DGCN.

In the training process, the goal is to minimize the error between the observed traffic flow states Y on the road network and the forecasted states $\hat{Y}$. Thus, the loss function can be defined as

$$loss = \frac{1}{2} \| \hat{Y} - Y \|_2 + \delta L_{reg} \quad (16)$$

where δ is a weighing factor, and $L_{reg} = \sum_{i=1}^{N_\theta} \theta_i^2$ represent the L_2 regularization term of all learnable parameters θ_i , which has the function of preventing the overfitting problem.

4. Experimental Comparisons on a Real-Life Dataset

This section aims to verify the effectiveness and superiority of the proposed T-DGCN model by performing comparative experiments on real-life datasets. In Section 4.1, we describe the utilized traffic dataset, including information on the moving velocity and turning directions at intersections, on the road network of Shenzhen, China. Section 4.2 introduces the baseline methods and evaluation metrics in the experimental comparisons. Finally, the experimental results are presented to demonstrate the superior performance of the proposed model in Section 4.3.

4.1. The Description of the Real-Life Dataset

There have been various traffic flow datasets, such as the PeMSD and METR-LA [39], designed for the performance evaluation of distinct forecasting models. However, they are mostly collected by fixed sensors on road segments, which lack the turning direction information of vehicles at intersections and cannot support directed graph construction. In recent years, GPS-equipped taxicabs have been employed as mobile sensors to constantly monitor the traffic rhythm of a city and to record the turning directions of taxis on road networks [40]. In China, Shenzhen city has more than 16,000 taxis that operate on the road network [41], and relevant studies have confirmed the ability of these taxi trajectories to reflect real traffic flow states on road networks [42]. Thus, we built a new large-scale traffic dataset based on the taxi trajectories of Shenzhen. The original dataset was downloaded from the Shenzhen Municipal Government Data Open Platform [43], which contains approximately 1 billion taxi trajectory points from 1–31 January 2012, which include multiple attribute information, such as taxi IDs, spatial locations, timestamps, and instantaneous velocities. For any road segment in any time interval, this study utilizes the average velocities of vehicles every 15 min on each road segment to represent the velocity of traffic flow. Figure 4 shows the spatial distribution of the road network in the study, which includes 672 interconnected road segments in major districts of Shenzhen.

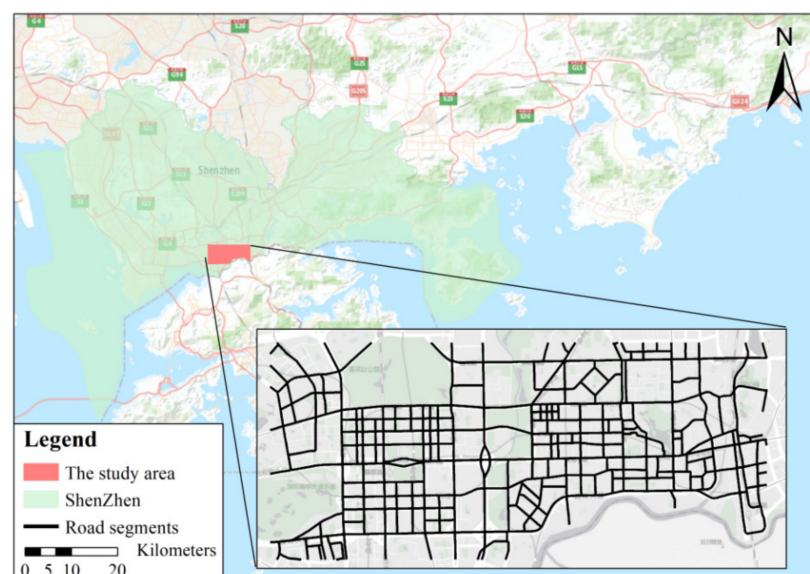


Figure 4. Spatial distribution of the road network in the major districts of Shenzhen city.

In the experiments, to obtain faster convergence, we normalized all the input velocity values to 0–1. According to chronological order, the first 60% of the whole dataset is used as the training set, while the following 20% and the last 20% are utilized for validation and testing, respectively.

4.2. Baseline Methods and Evaluation Metrics

To verify the superiority of the proposed traffic flow forecasting method, a total of seven representative models, namely the historical average (HA) model [44], ARIMA [16], the vector auto-regression (VAR) model [45], the support vector regression (SVR) model [46], the fully connected GRU (FC-GRU) model [27], the temporal graph convolutional network (T-GCN) model [2], and the diffusion convolutional recurrent neural network (DCRNN) model [13], were selected as the baseline methods to implement experimental comparisons with the proposed model. The first four models are traditional machine learning-based methods, while the last three models were designed by modifying and integrating state-of-the-art deep neural networks.

In addition, the following three quantitative metrics were used to conduct the accuracy assessment of the traffic forecasting results obtained by different methods, including the root mean squared error $RMSE = \sqrt{\frac{1}{2} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$, the mean absolute error $MAE = \frac{1}{2} \sum_{i=1}^n |y_i - \hat{y}_i|$, and the accuracy $AC = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)}{\sum_{i=1}^n (y_i - \bar{y})}$, where y_i and $\hat{y}_i$ represent the observed and forecasted values of the traffic flow velocity, respectively, while $\bar{y}$ denotes the average observations. RMSE and MAE were both utilized to measure forecasting errors, while AC indicated the forecasting precision. Therefore, high forecasting accuracies correspond to smaller RMSE and MAE values and larger AC values.

4.3. Comparative Analysis of the Experimental Results

In the experiments, we aimed to forecast the traffic flow velocity on road segments by using the proposed method and the baseline methods introduced in Section 4.2. The parameters included in the baseline methods were determined by referring to the identical criterion used in original articles or related articles. Specifically, the orders were set to (3, 0, 1) in the ARIMA model. In the VAR model, the lag was set to 3. The penalty term and the number of historical observations in the SVR model were set to 0.1 and 12, respectively. For the FC-GRU and T-GCN models, we set the number of hidden units to be 100.

Regarding the proposed method, we selected the appropriate parameters by comparing the forecasting performance of the candidates on the validation set. Specifically, we designed 16 hidden units in the directed graph convolution layers. For the keyframe sequence, the length of the tendency-related sequence and the time bandwidth of the periodicity-related sequence were set to $tl = 12$ and $tw = 5$, respectively, and the number of cycles was set to $N_p = 3$. In the transformer structure, we set the dimensions of the subspaces as $d_k = 8$ and $d_v = 16$, while the numbers of cells in the encoder and decoder layers were both set to 3. Additionally, to simultaneously learn the short- and long-term temporal dependence, the number of single-head attention nodes was set to be $N_h = 2$. In the training phase, we set up a batch size of 64 and 1000 epochs, while the learning rate was initialized as 0.0001 and was halved when the RMSE values remained unchanged for two epochs. All of the hyperparameters are classified and listed in Table 1. The proposed T-DGCN model was optimized using adaptive moment estimation (Adam) [47] and was implemented based on the PyTorch framework [48].

Table 1. The hyperparameters of the T-DGCN and the training process.

Hyperparameters of the T-DGCN								Hyperparameters of Training		
d_{model}	tl	tw	N_p	d_k	d_v	N_{cell}	N_h	Batch Size	Learning Rate	Epochs
16	12	5	3	8	16	3	2	64	0.0001	1000

Table 2 presents the quantitative evaluation results of the forecasted values obtained by different methods on the traffic flow data from the road network of Shenzhen. It is obvious that the deep neural network (i.e., T-GCN)-based models have significantly higher forecasting accuracy than the classical machine learning-based methods (i.e., ARIMA, VAR, and SVR). It can be concluded that deep neural networks have advantages in capturing the nonlinear features related to spatiotemporal dependence. Note that the T-GCN model has traffic forecasting performance similar to that of the FC-GRU model regardless of the forecasting step length. This illustrates that the topology-based undirected graph convolution operator has limits in modeling the spatiotemporal evolution of traffic flow.

Table 2. Quantitative evaluation results of different methods in traffic flow forecasting.

Step Size	Metric	The Proposed and the Baseline Methods							
		HA	ARIMA	VAR	SVR	FC-GRU	T-GCN	DCRNN	T-DGCN
1 (15 min)	RMSE	9.62	8.91	7.66	8.74	4.85	4.85	4.91	4.56 ¹
	MAE	6.11	5.62	4.88	5.34	2.95	3.40	3.22	2.97 ¹
	AC	0.69	0.71	0.74	0.70	0.81	0.82	0.81	0.85 ¹
2 (30 min)	RMSE	9.77	8.85	7.26	9.13	4.95	4.89	5.03	4.64 ¹
	MAE	6.25	5.53	4.86	5.69	3.10	3.43	3.35	3.03 ¹
	AC	0.69	0.72	0.78	0.69	0.79	0.80	0.81	0.82 ¹
3 (45 min)	RMSE	9.90	8.22	7.47	9.38	5.05	4.99	5.13	4.67 ¹
	MAE	6.37	5.63	5.03	5.91	3.20	3.10	3.57	3.09 ¹
	AC	0.68	0.74	0.77	0.68	0.80	0.79	0.79	0.81 ¹

¹ Black bold font indicates the best performance.

The proposed T-DGCN model outperforms all seven baseline methods in terms of the three evaluation metrics for different step sizes. More specifically, the forecasting results of the proposed directed graph convolution-based method yield smaller RMSE and MAE values and larger AC values than the other two current deep neural network-based methods (i.e., FC-GRU and T-GCN). For example, for traffic flow forecasting in 15 min, the RMSE value of the proposed T-DGCN model is approximately 6% lower than that of the T-GCN model, while the AC value is approximately 6% higher. For the forecasting step sizes of 30 min and 45 min, the proposed method outperforms both FC-GRU and T-GCN in terms of all three metrics, to a large degree confirms the stable performance of the proposed method to a large degree.

Furthermore, we specifically selected two road segments and visualized the results forecasted by the proposed method. The T-GCN model, which shows the best performance of the seven baseline methods, was selected as the representative for the comparisons. As shown in Figure 5, both models fit the curve of the observed traffic flow time series well. In detail, the T-GCN generates smoother forecasted results than the T-DGCN, which means that the curves produced by the T-DGCN contain more high-frequency components. In other words, the T-DGCN has obvious advantages in capturing drastic variations in traffic flow velocities.

In addition to the forecasting accuracy, comparative experiments were further conducted on the computational efficiency of both the baseline and the proposed methods. We ran all of the models on a computer with 128 G memory and 16 CPU cores at 2.9 GHz. Table 3 provides the efficiency evaluation results of different methods. One can see that all of the models have the capacity of outputting one-step forecasting results within 4 s. In other words, the computational time of all of the models can meet the requirements of real-time traffic flow forecasting given different forecasting steps (i.e., 15 min, 30 min, and 45 min). For deep learning-based methods, the running time on another computer with a Nvidia RTX3090 GPU indicates that the computation speed can be increased by nearly 10 times. In summary, the proposed method can achieve the highest forecasting accuracy within an acceptable computational time.

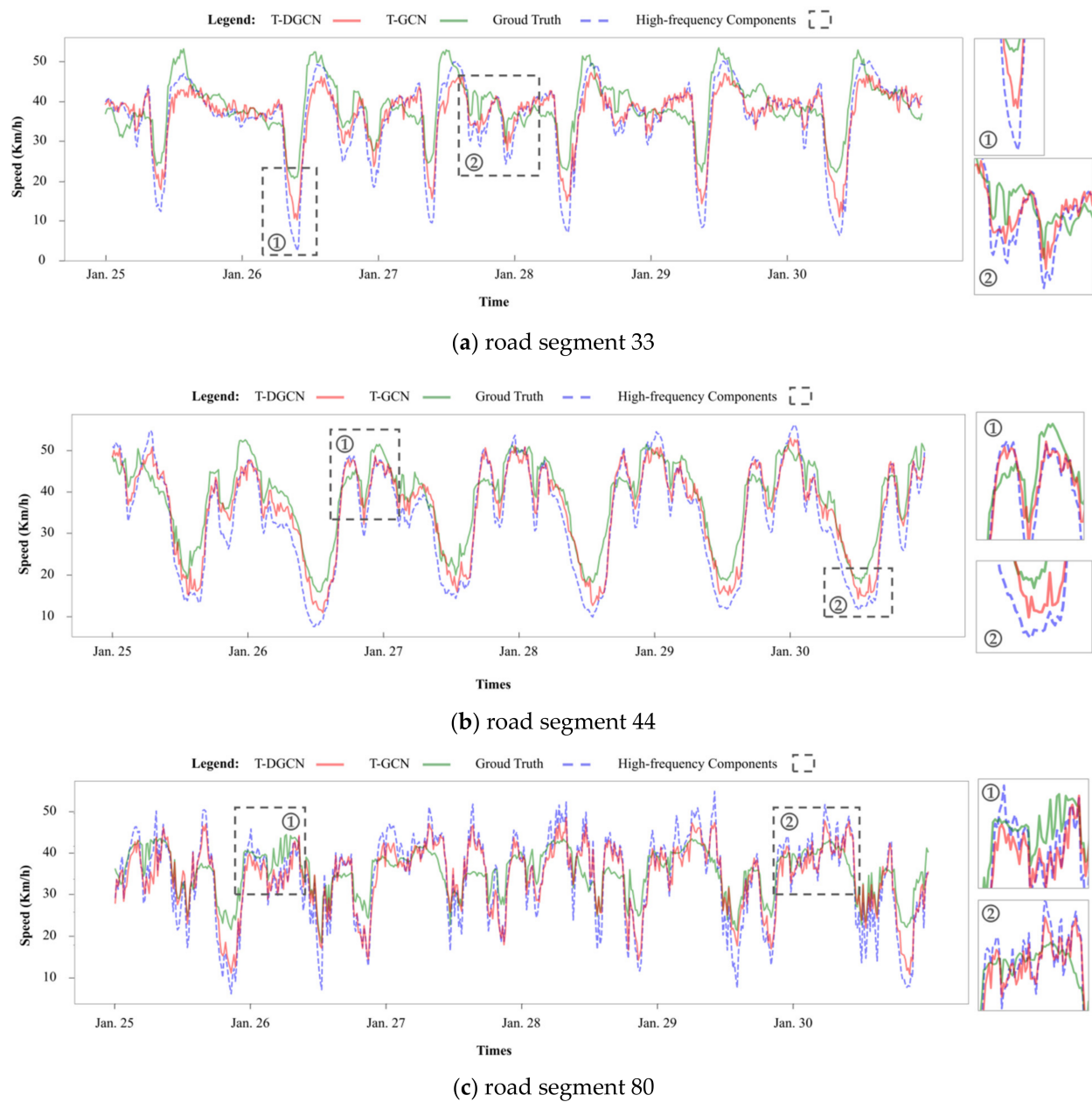


Figure 5. The forecasting results of the traffic flow time series on (a) road segment 33, (b) road segment 44, and (c) road segment 80 by the T-GCN and the proposed T-DGCN models with time resolution of 15 min.

Table 3. Efficiency evaluation results of different methods in traffic flow forecasting.

Device	The Proposed and the Baseline Methods							
	HA	ARIMA	VAR	SVR	FC-GRU	T-GCN	DCRNN	T-DGCN
CPU	<0.10 s	3.74 ± 0.06 s	3.56 ± 0.03 s	2.75 ± 0.05 s	2.16 ± 0.16 s	2.45 ± 0.19 s	3.78 ± 0.23 s	3.68 ± 0.30 s
GPU	/	/	/	/	0.17 ± 0.00 s	0.39 ± 0.00 s	0.59 ± 0.01 s	0.52 ± 0.01 s

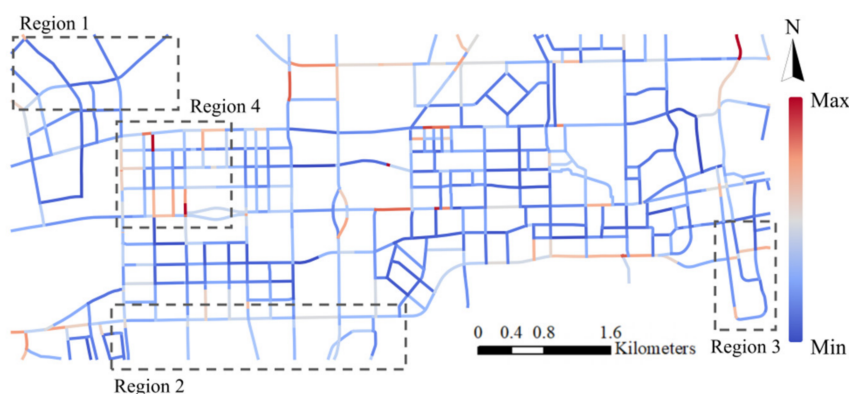
5. Discussion and Explanation of the Experimental Results

In this section, we further analyze the experimental results obtained by the proposed T-DGCN model from three aspects, namely the spatial distribution of the forecasting errors in Section 5.1, the temporal distribution of the forecasting errors (which refer to the RMSE values in the following sub-sections) in Section 5.2, and the multiterm temporal dependence in Section 5.3. Based on the analysis in the above subsections, we will provide the discussion

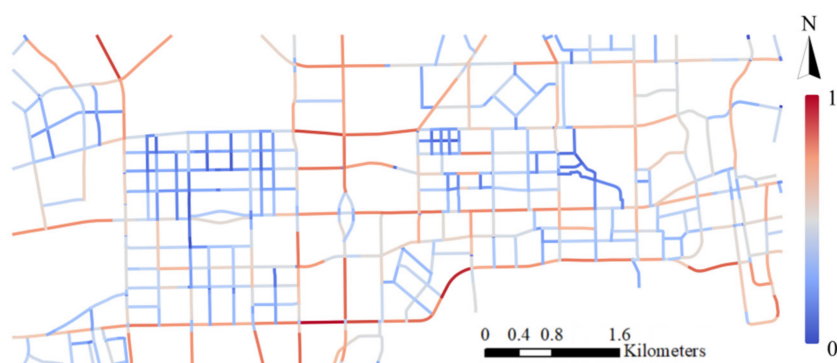
in Section 5.4. The purpose of this section is to provide convincing explanations for the superior performance of the proposed method.

5.1. The Spatial Distribution of the Forecasting Errors

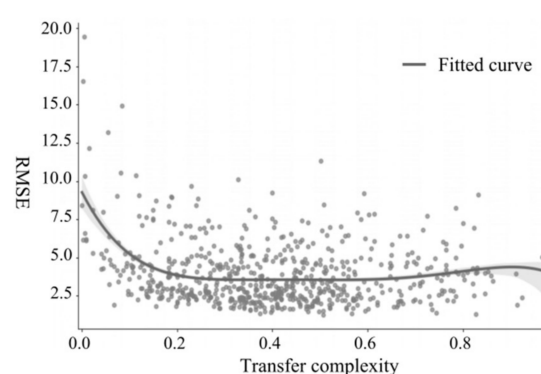
Figure 6a visualizes how the forecasting errors are distributed on the road network. Overall, the proposed T-DGCN model obtains forecasting results with small errors on most road segments. To analyze the relationship between the forecasting errors and the vehicle movement patterns, the transfer complexity value tc_i is calculated by $tc_i = \text{norm}(\sum_{j=1, j \neq i}^{672} 1\{\gamma_{ij}\})$, where indicator function $1\{\cdot\}$ enumerates the frequency of nonzero γ values and where $\text{norm}(\cdot)$ denotes a min-max normalization function. The distribution of tc_i is illustrated in Figure 6b. Visual comparisons show that the forecasting errors have negative correlations with the transfer complexity values. Taking the four highlighted regions in Figure 6a as examples, the road segments in Regions 1–3, which are located at the edge of the study area, contain incomplete topological structures but have high transfer complexity values and small forecasting errors. In contrast, despite the rich topology information in the road segments of Region 4, the low transfer complexity values correspond to the low forecasting accuracies.



(a) The distributions of the forecasting errors



(b) The distributions of the transfer complexity



(c) Fitting relations

Figure 6. Relationships between the forecasting errors and transfer complexity values with time step of 15 min, where (a) and (b) indicate the distributions of the forecasting errors and the transfer complexity values on the road network, while (c) depicts their fitting relationships using a scatter plot.

Moreover, Figure 6c presents a fitted curve to depict the relationships between the transfer complexity values and forecasting errors in a more intuitive way. It can be observed that an approximately negative linear relationship exists in the case of transfer complexity values smaller than 0.2. When the transfer complexity values exceed 0.2, the forecasting accuracies remain at a higher level. Furthermore, Figure 7 visualizes the normalized

Laplacian matrices of the topology-based undirected graph and the proposed Markov-based directed graph. On the one hand, the Laplacian matrix of the directed graph contains more nonzero elements, which means that the graph convolution filter can aggregate more neighborhood information than the undirected graph structure. On the other hand, the variable values of the diagonal elements indicate that the self-influences receive more attention in the directed graph structure.

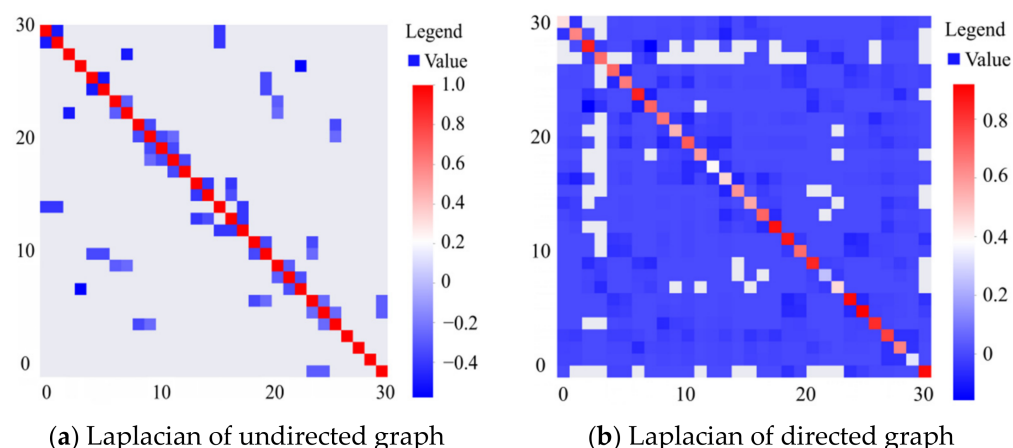


Figure 7. Visualization of the normalized Laplacian matrices of (a) the topology-based undirected graph and (b) the Markov-based directed graph for 30 selected road segments.

5.2. The Temporal Distribution of the Forecasting Errors

Figure 8 displays the average hourly distribution of the forecasting errors obtained by implementing the proposed method on the testing set. The T-DGCN has the ability to limit the forecasting errors to approximately four in the majority of timestamps. Here, interestingly, the forecasting errors during 0:00–6:00, especially those between 3:00–6:00, are significantly higher than those during other time periods.

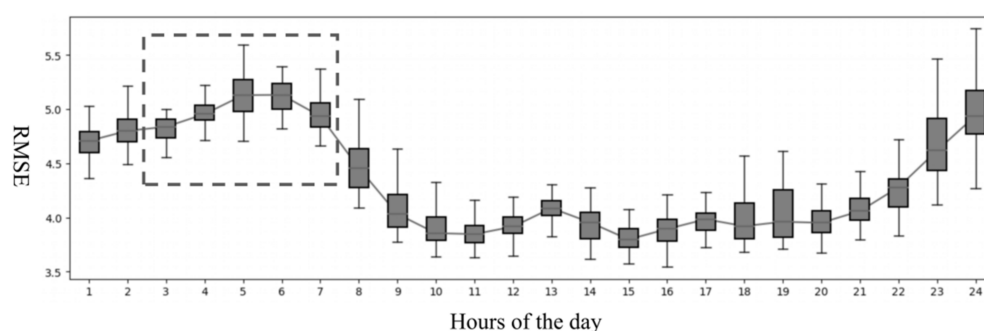


Figure 8. The average hourly distribution of the forecasting errors obtained by implementing the proposed method on the testing set, where the line represents the median of the forecasting errors.

This distribution characteristic is highly consistent with that described in a previous study [1]. The existing inferences suggest that this may be a result of the magnitude of traffic flow speed and the noise in records. However, Figure 9a,b illustrate the homogeneous distributions of the traffic flow velocities and standard deviation in a whole day, which rejects the above inferences. In this research, we further calculated the average hourly distribution of the number of vehicles in Figure 9c. Clearly, the average number of vehicles is very small during the early morning hours, which is in accordance with the distribution of the prediction forecasting errors.

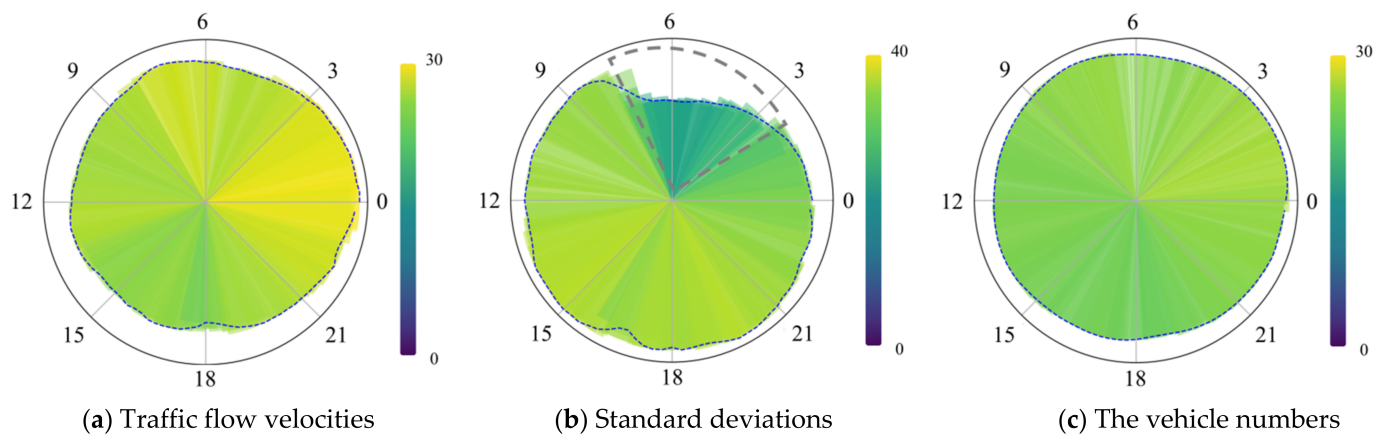


Figure 9. The average hourly distributions of (a) the traffic flow velocities, (b) the standard deviations of traffic flow velocities, and (c) the vehicle numbers.

5.3. Analysis of the Temporal Dependence

Figure 10 visualizes the multihead attention scores of four forecasting cases in the transformer structure. The scores quantify the contribution degree of the observations in the keyframe sequence to the traffic flow states to be forecasted. With the number of single-head attention nodes set to two, the training process automatically differentiates the two attention heads. The two attention heads learn the short-term dependence (i.e., the tendency) and the long-term dependence (i.e., the periodicity) of traffic flow. Specifically, Head-2 in Case 1 has higher attention scores in the beginning parts of the tendency-related sequence, while the ending parts make more contributions to the forecasted states in Case 4. For Cases 2 and 3, the middle parts in the tendency-related sequence are considered to be more important than the beginning and ending parts by the transformer structure. In addition, the heterogeneity of long-term dependence is adaptively captured, as reflected by the distributions of attention scores in the periodicity-related sequence of Head-1.

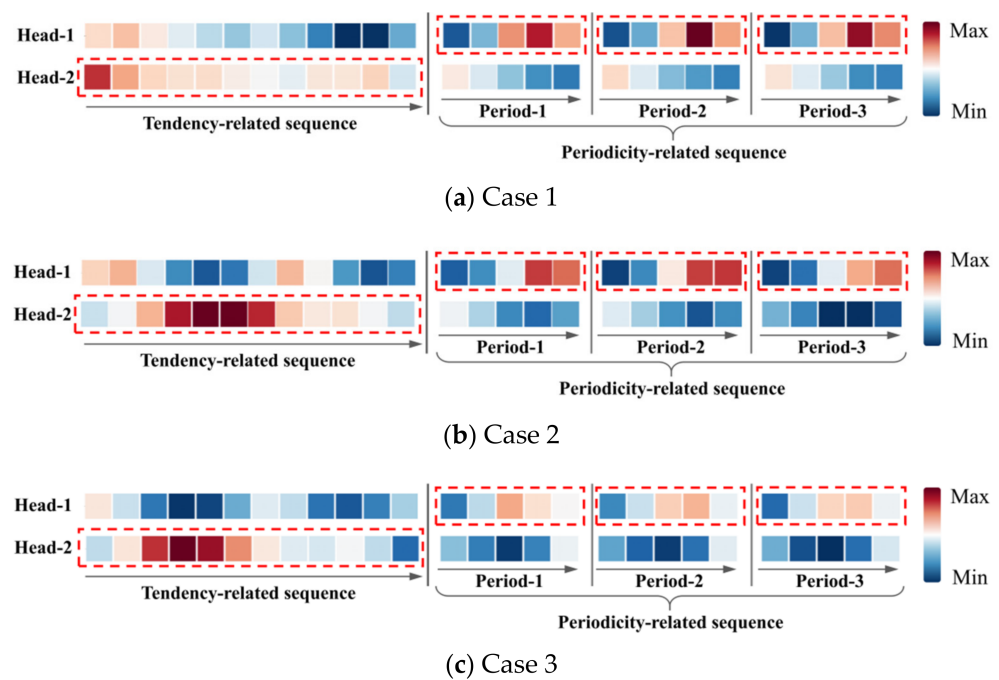


Figure 10. Cont.

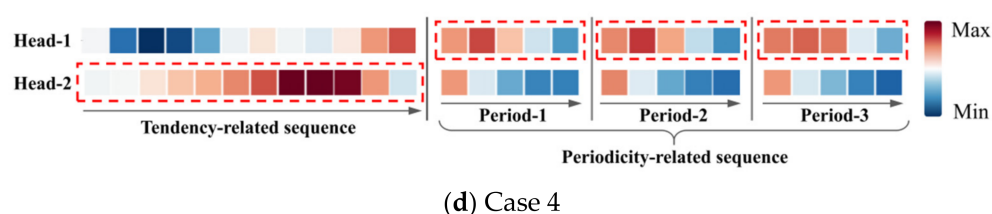


Figure 10. Visualization of the two-head attention scores regarding (a) Case 1, (b) Case 2, (c) Case 3, and (d) Case 4 in the transformer structure.

Furthermore, we utilized the auto-correlation function (ACF) to demonstrate the rationality and effectiveness of the trained two-head attention. Figure 11a shows the calculated autocorrelation coefficients of the original traffic flow time series with different time lags, where each line describes the autocorrelation for each road segment. It is obvious that the utilized traffic flow data contain significant tendencies and periodicities that appear to be discrepant between road segments. Moreover, Figure 11b depicts the relationship between the autocorrelation coefficients and forecasting errors. The results indicate that the forecasting errors of the proposed method stabilize at low levels for road segments with average autocorrelation coefficients larger than 0.2.

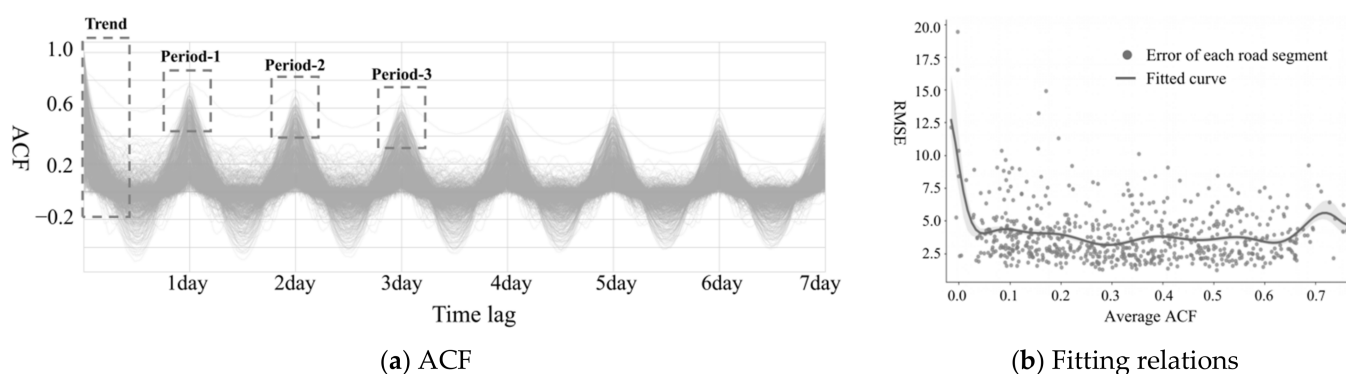


Figure 11. (a) The autocorrelation coefficients of the utilized traffic flow time series; (b) the relationships between the average autocorrelation coefficients and the forecasting errors.

5.4. Discussion

Through the above analysis of the experimental results, we are able to provide a comprehensive discussion regarding the outperformance of the proposed method in terms of the accuracy of traffic flow forecasting from the following three aspects.

In the spatial dimension, the directed graph structure enables the neural network to leverage more associated information with the help of the Markov transfer matrix, which is a critical factor in higher traffic flow forecasting accuracies. In the temporal dimension, the multihead attention in the proposed method has the ability to adaptively learn the short-term and long-term temporal dependence of traffic flow states observed on different road segments at distinct timestamps. Based on the above two factors, we can make convincing arguments that the proposed method is superior to the baseline methods.

Furthermore, in real-world applications, the sparse observations of traffic flow states in the early morning hours may increase the unreliability of space–time dependency feature learning and the associated forecasting errors. In other words, the proposed model performs better when there are more vehicles on the road network. However, traffic forecasting is more important and needed during peak hours to serve as many vehicles as possible, which is also the period with the highest forecasting accuracy of the proposed method. Hence, the T-DGCN model is able to meet the needs of realistic traffic forecasting tasks.

6. Conclusions

This study designed a new method called the temporal directed graph convolution Network (T-DGCN) to achieve high-precision traffic flow forecasting by adaptively capturing complicated spatial and temporal dependence. Specifically, in the spatial dimension, the idea of Markov chains is introduced to construct a directed graph for a road network by taking the vehicle turning behaviors at intersections into account. On this basis, we employed a directed graph convolution operator to learn spatial dependence features. In the time dimension, we built a keyframe sequence for each forecasted state and used the transformer structure to capture both short-term and long-term temporal dependence. In the experiments, real-world taxi trajectory points in Shenzhen city, China, were utilized to estimate historical traffic flow states on the road network to perform experimental comparisons between the proposed method and seven commonly used representative baseline methods using different evaluation metrics. The experimental results demonstrate the superiority of the proposed method in terms of traffic flow forecasting accuracy. In addition, we further discussed the forecasting results obtained by the proposed method from the space–time distributions of the forecasting errors and the multiterm temporal dependence. To a large extent, the discussions rationalize the high forecasting accuracy of the proposed method.

In the future, we will pay attention to the following three aspects of published works: The first is to make comparisons between the performance of model-based traffic simulators and deep learning models in real-time traffic flow forecasting. The second is to investigate the impacts of incompleteness of traffic flow data on the model training process and on measuring the uncertainty degree of forecasting results by leveraging statistical models. Third, focus will be given to generalize the proposed T-DGCN model to improve its applications in diverse traffic scenarios.

Author Contributions: Conceptualization, Kaiqi Chen and Min Deng; methodology, Kaiqi Chen; software, Kaiqi Chen; validation, Kaiqi Chen and Yan Shi; formal analysis, Kaiqi Chen and Yan Shi; resources, Min Deng; data curation, Kaiqi Chen; writing—original draft preparation, Kaiqi Chen; writing—review and editing, Yan Shi; visualization, Kaiqi Chen; supervision, Min Deng and Yan Shi; project administration, Min Deng, Kaiqi Chen and Yan Shi; funding acquisition, Min Deng. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported by the National Natural Science Foundation of China (NSFC), Project No. 41730105 and 42071452; the Fundamental Research Funds for the Central Universities of Central South University, Project No. 2020zzts687; and the Natural Science Foundation of Hunan Province, China, Project No. 2020JJ4696.

Data Availability Statement: As the data also form part of an ongoing study, the raw data cannot be shared at this time.

Acknowledgments: The authors would like to thank the reviewers for their useful comments and suggestions for this paper. This work was supported by the National Natural Science Foundation of China (NSFC), Project No. 41730105 and 42071452; the Fundamental Research Funds for the Central Universities of Central South University, Project No. 2020zzts687; and the Natural Science Foundation of Hunan Province, China, Project No. 2020JJ4696.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Y.; Cheng, T.; Ren, Y.; Xie, K. A novel residual graph convolution deep learning model for short-term network-based traffic forecasting. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 969–995. [\[CrossRef\]](#)
2. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [\[CrossRef\]](#)
3. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Lemus-Zúñiga, L.G.; Cano, J.C.; Manzoni, P.; Gayraud, T. A centralized route-management solution for autonomous vehicles in urban areas. *Electronics* **2019**, *8*, 722. [\[CrossRef\]](#)

4. Chen, C.; Li, K.; Teo, S.G.; Zou, X.; Wang, K.; Wang, J.; Zeng, Z. Gated residual recurrent graph neural networks for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 485–492. [\[CrossRef\]](#)
5. Li, Y.; Shahabi, C. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Spec.* **2018**, *10*, 3–9. [\[CrossRef\]](#)
6. Pan, Z.; Liang, Y.; Wang, W.; Yu, Y.; Zheng, Y.; Zhang, J. Urban traffic prediction from spatio-temporal data using deep meta learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; ACM: New York, NY, USA, 2019; pp. 1720–1730.
7. Saidallah, M.; El Fergougui, A.; Elalaoui, A.E. A comparative study of urban road traffic simulators. *MATEC Web Conf.* **2016**, *81*, 05002. [\[CrossRef\]](#)
8. Chao, Q.; Bi, H.; Li, W.; Mao, T.; Wang, Z.; Lin, M.C.; Deng, Z. A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. *Comput. Graph. Forum* **2020**, *39*, 287–308. [\[CrossRef\]](#)
9. Sewall, J.; Van Den Berg, J.; Lin, M.; Manocha, D. Virtualized traffic: Reconstructing traffic flows from discrete spatiotemporal data. *IEEE Trans. Vis. Comput. Graph.* **2011**, *17*, 26–37. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Duan, P.; Mao, G.; Zhang, C.; Wang, S. STARIMA-based traffic prediction with time-varying lags. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1610–1615.
11. Bogaerts, T.; Masegosa, A.D.; Angarita-Zapata, J.S.; Onieva, E.; Hellinckx, P. A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transp. Res. C Emerg. Technol.* **2020**, *112*, 62–77. [\[CrossRef\]](#)
12. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
13. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
14. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
15. Drew, D.R. *Traffic Flow Theory and Control*; McGraw-Hill: New York, NY, USA, 1968.
16. Lee, S.; Fambro, D.B. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transp. Res. Rec.* **1999**, *1678*, 179–188. [\[CrossRef\]](#)
17. Hamed, M.M.; Al-Masaeid, H.R.; Said, Z.M.B. Short-term prediction of traffic volume in urban arterials. *J. Transp. Eng.* **1995**, *121*, 249–254. [\[CrossRef\]](#)
18. Van Der Voort, M.; Dougherty, M.; Watson, S. Combining kohonen maps with arima time series models to forecast traffic flow. *Transp. Res. C Emerg. Technol.* **1996**, *4*, 307–318. [\[CrossRef\]](#)
19. Guo, J.; Huang, W.; Williams, B.M. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. C Emerg. Technol.* **2014**, *43*, 50–64. [\[CrossRef\]](#)
20. Vanajakshi, L.; Rilett, L.R. A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed. In Proceedings of the IEEE Intelligent Vehicles Symposium 2004, Parma, Italy, 14–17 June 2004; pp. 194–199. [\[CrossRef\]](#)
21. Su, H.; Zhang, L.; Yu, S. Short-term traffic flow prediction based on incremental support vector regression. In Proceedings of the Third International Conference on Natural Computation (ICNC 2007), Haikou, China, 24–27 August 2007; pp. 640–645.
22. Gopi, G.; Dauwels, J.; Asif, M.T.; Ashwin, S.; Mitrovic, N.; Rasheed, U.; Jaillet, P. Bayesian support vector regression for traffic speed prediction with error bars. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems, The Hague, The Netherlands, 6–9 October 2013; pp. 136–141.
23. Yin, H.; Wong, S.C.; Xu, J.; Wong, C.K. Urban traffic flow prediction using a fuzzy-neural approach. *Transp. Res. C Emerg. Technol.* **2002**, *10*, 85–98. [\[CrossRef\]](#)
24. Cai, P.; Wang, Y.; Lu, G.; Chen, P.; Ding, C.; Sun, J. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transp. Res. C Emerg. Technol.* **2016**, *62*, 21–34. [\[CrossRef\]](#)
25. Sun, S.; Zhang, C.; Yu, G. A bayesian network approach to traffic flow forecasting. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 124–132. [\[CrossRef\]](#)
26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
27. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
28. Ma, X.; Yu, H.; Wang, Y.; Wang, Y. Large-scale transportation network congestion evolution prediction using deep learning theory. *PLoS ONE* **2015**, *10*, e0119044. [\[CrossRef\]](#)
29. Tian, Y.; Li, P. Predicting short-term traffic flow by long short-term memory recurrent neural network. In Proceedings of the IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, China, 19–21 December 2015; pp. 153–158.
30. Wu, Y.; Tan, H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv* **2016**, arXiv:1612.01022.
31. Yu, B.; Li, M.; Zhang, J.; Zhu, Z. 3D graph convolutional networks with temporal graphs: A spatial information free framework for traffic forecasting. *arXiv* **2019**, arXiv:1903.00919.

32. Chen, Z.; Wen, J.; Geng, Y. Predicting future traffic using Hidden Markov Models. In Proceedings of the 2016 IEEE 24th International Conference on Network Protocols, Singapore, 8–11 November 2016; pp. 1–6.
33. Ma, Y.; Hao, J.; Yang, Y.; Li, H.; Jin, J.; Chen, G. Large-scale transportation network congestion evolution prediction using deep learning theory. *arXiv* **2019**, arXiv:1907.08990. [[CrossRef](#)]
34. Chung, F.R.; Graham, F.C. *Spectral Graph Theory*; American Mathematical Society: Providence, RI, USA, 1997.
35. Zhang, J.; Zheng, Y.; Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; ACM: New York, NY, USA, 2017; pp. 1655–1661.
36. Cai, L.; Janowicz, K.; Mai, G.; Yan, B.; Zhu, R. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Trans. GIS* **2020**, *24*, 736–755. [[CrossRef](#)]
37. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Stanford University: Stanford, CA, USA, 2016; pp. 855–864.
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
39. Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; Jia, Z. Freeway performance measurement system: Mining loop detector data. *Transp. Res. Rec.* **2001**, *1748*, 96–102. [[CrossRef](#)]
40. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; Huang, Y. T-drive: Driving directions based on taxi trajectories. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; ACM: New York, NY, USA, 2010; pp. 99–108.
41. Nie, Y. How can the taxi industry survive the tide of ridesourcing? Evidence from Shenzhen, China. *Transp. Res. C Emerg. Technol.* **2017**, *79*, 242–256. [[CrossRef](#)]
42. Castro, P.S.; Zhang, D.; Li, S. Urban traffic modelling and prediction using large scale taxi GPS traces. In *Pervasive Computing*; Kay, J., Lukowicz, P., Tokuda, H., Olivier, P., Krüger, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 57–72.
43. Shenzhen Municipal Government Data Open Platform. Available online: <https://opendata.sz.gov.cn/> (accessed on 16 June 2019).
44. Liu, J.; Guan, W. A summary of traffic flow forecasting methods. *J. Highw. Transp. Res. Dev.* **2004**, *3*, 82–85.
45. Schimbschi, F.; Moreira-Matias, L.; Nguyen, V.X.; Bailey, J. Topology-regularized universal vector autoregression for traffic forecasting in large urban areas. *Expert Syst. Appl.* **2017**, *82*, 301–316. [[CrossRef](#)]
46. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
47. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
48. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv* **2019**, arXiv:1912.01703.