sustainability

Article

Data-Driven Graph Filter-Based Graph Convolutional Neural Network Approach for Network-Level Multi-Step Traffic Prediction

Lei Lin [1], Weizi Li [2] and Lei Zhu [3,*]

[1] Goergen Institute for Data Science, 1209 Wegmans Hall, University of Rochester, Rochester, NY 14627, USA
[2] Department of Computer Science, The University of Memphis, Memphis, TN 38152, USA
[3] Systems Engineering & Engineering Management, The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223, USA
* Correspondence: lei.zhu@uncc.edu; Tel.: +1-(704)-687-6054

Abstract: Accurately predicting network-level traffic conditions has been identified as a critical need for smart and advanced transportation services. In recent decades, machine learning and artificial intelligence have been widely applied for traffic state, including traffic volume prediction. This paper proposes a novel deep learning model, Graph Convolutional Neural Network with Data-driven Graph Filter (GCNN-DDGF), for network-wide multi-step traffic volume prediction. More specifically, the proposed GCNN-DDGF model can automatically capture hidden spatiotemporal correlations between traffic detectors, and its sequence-to-sequence recurrent neural network architecture is able to further utilize temporal dependency from historical traffic flow data for multi-step prediction. The proposed model was tested in a network-wide hourly traffic volume dataset between 1 January 2018 and 30 June 2019 from 150 sensors in the Los Angeles area. Detailed experimental results illustrate that the proposed model outperforms the other five widely used deep learning and machine learning models in terms of computational efficiency and prediction accuracy. For instance, the GCNN-DDGF model improves MAE, MAPE, and RMSE by 25.33%, 20.45%, and 29.20% compared to the state-of-the-art models, such as Diffusion Convolution Recurrent Neural Network (DCRNN), which is widely accepted as a popular and effective deep learning model.

Keywords: deep learning; graph convolutional gated recurrent neural network; data-driven graph filter; networkwide multi-step traffic prediction

Citation: Lin, L.; Li, W.; Zhu, L. Data-Driven Graph Filter-Based Graph Convolutional Neural Network Approach for Network-Level Multi-Step Traffic Prediction. *Sustainability* **2022**, *14*, 16701. https://doi.org/10.3390/su142416701

Academic Editor: Sara Moridpour

Received: 28 October 2022
Accepted: 9 December 2022
Published: 13 December 2022

1. Introduction

Short-term traffic state prediction (such as traffic speed and flow prediction in the next 15 min) is a critical element in transportation control and operation in smart cities. Extensive studies on short-term traffic state prediction by using traditional statistical and machine learning models, such as the Autoregressive Integrated Moving Average model (ARIMA) [1,2], linear multi-regression dynamic model [3], k-Nearest Neighbor (k-NN) [4], Support Vector Machine [2,5], Artificial Neural Networks (ANN) [6,7] have been appearing in the recent decade. In addition, recent advances in deep learning models [8,9] have also been applied for short-term traffic state applications, and many studies have evidenced the perfect performance of deep learning models [10–13].

Although the progress of traffic state prediction models has been observed, previous studies could be improved from different aspects. From the spatial perspective, many existing studies only predict the traffic state of individual road links, intersections, and single or few traffic detectors [1–4,6,7,11]. For example, travel speed data were collected from two locations in a major ring road around Beijing by Ma et al. 2015 [11]. Therefore, the correlation between different sensors cannot be described by using those approaches. On the other hand, the majority of previous studies only makes single-step prediction for different next-time intervals, such as 15-min or one hour [1–5,7,10,11,13,14]. For instance, in a bike-sharing network study [10], the 22,304 hourly bike-sharing demand values are used to train the models. In comparison, the following 2000 hourly bike-sharing demand values are included in the validation dataset, and the remaining 2000 are taken as the testing dataset.

Several scholars have pointed out the necessity of multi-step traffic state prediction for the whole network [12,15]. For example, a single-step traffic state prediction for a fixed road segment is inadequate for some ITS applications, like dynamic travel route optimization. Recurrent neural network (RNN) is one of the most popular deep learning architectures, which is especially capable of modeling non-linear temporal dependencies in sequential data and has been applied in many recent traffic state prediction studies [11,14,16] for multiple-step predictions. The sequence-to-sequence recurrent neural network (Seq2Seq RNN) architecture was first proposed for natural language processing [17], and it is made up of two RNNs: the encoder and the decoder. The encoder takes a sequence of historical traffic states as the input, and the decoder generates an output sequence as predictions. Some previous studies train different deep learning models for various prediction horizons to realize the multi-step prediction [12,16,18,19]. However, training deep learning models for various prediction tasks is time-consuming, which limits their usage in practice. On the other hand, the wide adoption of traffic monitoring and sensing technologies, including loop detectors [20,21], Global Positioning System (GPS) [22,23], and remote traffic microwave sensors [11], as well as the emerging connected and automated vehicle (CAV) technologies [24,25], generating a large amount of traffic data, which enable the network-wide multi-step traffic state prediction.

One of the keys to improving network-wide traffic state prediction is to understand the spatiotemporal correlation between traffic sensors in the network [10,12,13,26–28]. A novel deep-learning-based traffic flow prediction method was developed by considering the spatial and temporal correlations inherently [12]. Additionally, an extended time-series-based approach takes into account spatial and temporal interactions in a new manner, specialized to the context of road traffic was proposed by Min and Wynter [28]. One intuitive assumption is that the latest measures from spatially near traffic sensors are adequate for forecasting their traffic states in the future [13]. However, in some scenarios, this assumption may not hold. As shown in Figure 1a, two sensors monitor traffic flow in two directions. In that case, traffic congestion observed by one sensor in one direction does not imply that the other direction will be jammed. Some researchers propose to utilize network topology distance to capture spatiotemporal correlation among sensors [27,29]. Figure 1b illustrates an example in which this assumption is also questionable. When an on-ramp locates on the road segment between two sensors, their traffic states, e.g., traffic flow, could be different due to the on-ramp, which offers inflow traffic. Besides the cases mentioned above, when an arterial road is served as an alternative to a nearby highway, their traffic states may be strongly correlated. However, they are not considered topological neighbors [13].

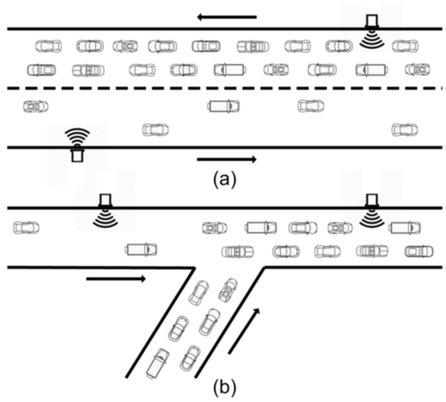

Figure 1. Freeway Segments with (**a**) two Spatially Near Traffic Sensors and (**b**) two Topologically Near Traffic Sensors.

The breakthrough of deep learning techniques, especially Convolutional Neural Networks (CNNs), has brought new insights to solve this problem. CNNs extract meaningful statistical patterns presented in image and video data [30]. The correlations among picture grids can be learned automatically through various localized filters. Because CNNs can only work on data with regular or well-defined structures, such as images, for network traffic state prediction, one method is geospatial map partition [26]. For instance, the whole city is converted into a grid map where each grid denotes a small region with a predefined and constant size. The pixel value of each grid can be calculated based on the inflow and outflow traffic through the small grid region. Although this approach can automatically capture the correlations among all grids, one drawback of the geospatial map partition approach is that it is hard to determine the size of the grid. A large grid may fail to satisfy the required granularity or resolution for specific traffic sensors or road segments, while a small grid will lead to a huge image matrix and increase the computational burden. Another image transformation approach is a time-space diagram [18], where the number of columns in the image is the length of time intervals, and the number of rows is the number of road segments or traffic sensors. At the same time, each pixel represents the average traffic speed for a road segment at the time interval. The issue with this approach is that the order or sequence of the row elements (e.g., road segments) is arbitrarily predefined. Hence, the correlations between road segments may not be fully captured by the localized kernel filters of CNNs.

Due to the limitation of CNNs in processing irregular network data (e.g., traffic state data from a road network), some scholars have extended CNNs to graph convolutional neural networks (GCNNs), which produce the convolution operations based on signal processing technologies and graph theory [30,31] and do not require the image transformation operations in network-wide traffic state prediction [27,29]. Instead, the input of the model is the traffic state data, which is directly embedded in a graph, where the nodes represent traffic sensors, and the adjacency matrix of the graph is constructed based on Euclidean distance [29] or road network distance [27]. GCNNs have outperformed traditional models as well as deep learning models in previous studies [27,29]. However, the assumption behind the CNN models still holds in GCNN models, which is the correlations between two sensors are stronger if they are spatially or topologically near than others. It makes little sense in some scenarios as described above, and it is expected the performance of the GCNNs can be further improved through a well-designed adjacency matrix [30,32].

Therefore, it is critical to accurately capture the spatiotemporal correlation between traffic sensors to model an irregular traffic network for network-wide traffic state forecasting. Additionally, computationally efficient and effective multiple-step traffic state forecasting is highly desired.

In this study, a novel GCNN with a data-driven graph filter (GCNN-DDGF) is proposed for network-wide traffic volume prediction. It has a few advantages compared with previous models. First, it keeps the merit of GCNN of applying to an irregular traffic network; furthermore, the data-driven graph filter (DDGF) enables it to automatically learn the adjacency matrix to capture the spatiotemporal correlation between traffic sensors instead of the predefinition, which may not be accurate. Second, instead of making merely single-step predictions, the GCNN-DDGF model is implemented on top of a sequence-to-sequence recurrent neural network (Seq2Seq RNN) architecture for efficient multi-step predictions by using the identical model to avoid a large amount of hyperparameter tuning. Additionally, the proposed model chooses the Gated Recurrent Unit (GRU) as the RNN cell, which has similar performance as the Long Short-term Memory Unit (LSTM) but is more computationally efficient [33]. The model also utilizes the fast approximation of the spectral graph convolution method from previous GCNN studies [32]. Both designs further speed up the training and modeling of the GCNN-DDGF model.

The experiments are conducted based on an hourly traffic volume dataset collected from 150 sensors from 1 January 2018 to 30 June 2019 in the Los Angeles area [21]. The GCNN-DDGF utilizes past 12-h traffic volumes to predict the next 12-h traffic volumes for all sensors. The results show that the proposed model outperforms two state-of-the-art deep learning models [27] and three traditional traffic volume prediction models in terms of criteria, including MAE, MAPE, and RMSE. The GCNN-DDGF has also demonstrated better performance from both the temporal and spatial aspects. Moreover, the training efficiency of the GCNN-DDGF is verified by experimental results.

The rest of the paper is organized as follows. The next section introduces relevant traffic state prediction studies. After that, the methodology part presents the proposed GCNN-DDGF model. The dataset and model results are then discussed in the Data and Experiment section. Finally, the last section closes the whole paper with conclusions and future research efforts.

2. Related Work

This section first introduces relevant studies considering spatial and temporal correlations in short-term traffic state predictions. Following that, models for network-wide and multi-step traffic state predictions will be presented separately.

2.1. Spatial and Temporal Correlations

It has been widely demonstrated that spatial and temporal correlations exist in traffic state data [12,13,28,34–36]. Traffic state data from a fixed sensor can be considered as a cyclical time series, which is usually correlated with the series collected from a nearby traffic sensor. The spatial and temporal correlations can be easily verified through the cross-correlation function [35] and the Hurst Exponent [2,34]. However, it is difficult to identify the fittest spatial and temporal features in the modeling process, e.g., determining relevant traffic state series from all sensors and the appropriate time lag for each series. Other studies take statistical criteria such as Akaike Information Criterion (AIC) and autocorrelation functions to identify relevant time lags in historical traffic state data [35,36]., while some studies rely on empirical experiments, e.g., grid search [12] or genetic algorithm (GA) [34], to determine the optimal number of previous time intervals to obtain time features. In terms of spatially correlated sensor identification, a few studies directly choose the same link sensors [34,35] or those at the adjacent upstream and downstream segments [36]. To predict traffic flow at a target sensor, Polson and Sokolov (2017) applied the LASSO regression analysis to select important predictors from six lagged measurements from 20 traffic sensors at the same link [13]. For network-wide traffic state prediction, it is unrealistic to identify the related traffic sensors for each of the hundreds of sensors in the road network using this approach.

2.2. Network-Wide Traffic State Prediction

Min and Wynter (2011) proposed one of the earliest network-wide traffic state prediction studies [28]. They applied a multivariate spatiotemporal autoregressive model

(MSTAR) to predict the network-wide speed and volume, considering neighboring links' effect on each link's traffic prediction. The adjacent links of a given link are predefined based on whether they are reachable with historical average link speed during a specified period. Moreover, Cai et al. (2016) implemented an enhanced k-NN model to predict the traffic state for 30 road segments [15]. The nearest neighbors are selected based on a composite equivalent distance that considers the physical distance, the correlation coefficient, and the connective grade between road segments.

Compared with the traditional modeling approaches that require well-designed temporal and spatial features [13,15,28,34–36], deep learning models are becoming more and more prevalent in network-wide traffic prediction. One of the main reasons is the representation learning capability that needs little effort to extract features from raw data manually [37]. Lv et al. (2015) built a Stacked Autoencoder (SAE) network that takes historical network-wide traffic flow data as a flat input vector [12]. They argue that SAE can capture inherent spatial and temporal correlations in data by itself. Some other robust deep learning architectures, such as CNNs, have emerged as a preferred choice because localized kernels are designed to capture relationships among regular grids in image data automatically. However, traffic state data have to be transferred into an image for CNNs [18,26,38]. GCNNs extended CNNs by considering spatial correlations between traffic sensors through the predefinition of the graph adjacency matrix, which is essentially a function of the spatial Euclidean distance or the topological road network distance [27,29].

2.3. Multi-Step Traffic State Prediction

Three approaches have been discovered in multi-step traffic state previous studies. The most straightforward approach is to make predictions directly for multiple steps [15,38]. Cai et al. (2016) built an enhanced k-NN model based on spatiotemporal state matrices, the output of which is also a matrix that represents the future 12-step prediction for the whole 30 road segments [15]. Wang et al. (2020) attempted to change the dimension of the output layer for CNNs to match the number of prediction steps [38]. In the second approach first builds single-step prediction models and provides prediction values, which are used as the observed true values for the following prediction steps. The whole process is repeated until the required prediction steps are reached [28,35,39]. The last approach builds isolated prediction models for various horizons. As representative examples, Lv et al. (2015) and Ma et al. (2017) trained multiple deep-learning models with various prediction horizons [12,18]. The hyperparameter tuning of these models is very time-consuming due to the complexity of the models.

As a prevalent deep learning architecture for sequential data generation and prediction, Seq2Seq RNN has been mainly applied for natural languages processing tasks such as machine translation [17] and informative conversation generation [40]. Seq2Seq RNN only needs to be trained once for multi-step prediction. Similar to the second approach, the decoder in Seq2Seq RNN takes old forecasts as the input to predict the following steps until the required number is satisfied [27].

3. Methodology

This section introduces the Graph Convolutional Neural Network with Data-driven Graph Filter (GCNN-DDGF) model from three components: encoder-decoder recurrent neural network architecture, spectral graph convolution, and data-driven graph filter.

3.1. Encoder-Decoder Recurrent Neural Network Architecture

Assuming a transportation road network has N sensors, the traffic volume of all sensors in hour k are $x_k \in \mathbb{R}^N$. The main task is to use the previous T-step traffic volumes $X_k \in \mathbb{R}^{N \times T}$, $X_k = [x_{k-T+1}, \ldots, x_k]$ to predict the next F-step traffic volumes $Y_k \in \mathbb{R}^{N \times F}$, $Y_k = [\hat{y}_{k+1}, \ldots, \hat{y}_{k+F}]$, which are the outputs of the model. To capture the temporal dependency, the GCNN-DDGF model implements the encoder-decoder recurrent neural network (RNN) architecture, in Figure 2.

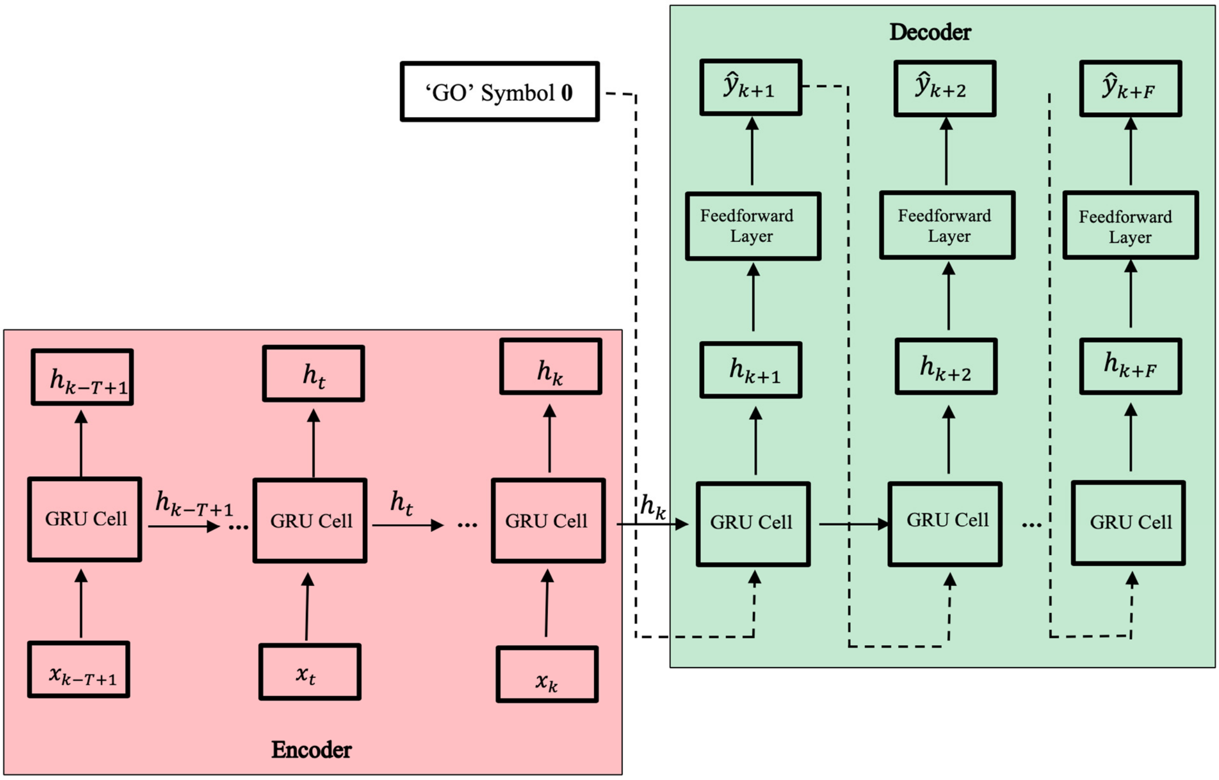

Figure 2. Encoder-Decoder Recurrent Neural Network Architecture.

The RNN cells include the Long Short-term Memory cell (LSTM) and Gated Recurrent Unit (GRU) cell, which both rely on the gate mechanism to modulate the flow of information inside the cell. In this study, GRU cells are used because of their simple structure, which only contains an update gate and a reset gate and can lead to more computational efficiency. As shown in the diagram, for each GRU cell in the encoder part, the input is x_t, $t = k - T + 1, \ldots, k$, and the update gate z_t and the reset gate r_t can be calculated as follows:

$$z_t = \sigma([h_{t-1}, x_t] \cdot W_z) \tag{1}$$

$$r_t = \sigma([h_{t-1}, x_t] \cdot W_r) \tag{2}$$

The corresponding output is the hidden state h_t, which can be calculated as follows:

$$\widetilde{h}_t = tahn([r_t \cdot h_{t-1}, x_t] \cdot W) \tag{3}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \widetilde{h}_t \tag{4}$$

Note σ and *tahn* represent the Sigmoid and Tangent activation functions. W_z, W_r, and W denote trainable weight parameters. The reset gate r_t determines the amount of information to keep from the previously hidden state h_{t-1}. The update gate z_t determines the information discarded from the previous step and how much information is to be added from the current time step.

In the decoder part, $Y_k = [\hat{y}_{k+1}, \hat{y}_{k+2}, \ldots, \hat{y}_{k+F}]$ represents the prediction results. h_k from the encoder is fed into the first GRU cell to initialize the hidden state. In the decoder, the input to the first GRU cell is a unique 'GO' symbol, which is a zero vector 0. Each of the rest of the GRU cells in the decoder takes the prediction of the preceding step as the input. The calculation of h_d, $d = k + 1, k + 2, \ldots, k + F$ is similar, as described in Equations (1)–(4). It is fed into a feedforward layer to generate the prediction $\hat{y}_d$:

$$\hat{y}_d = h_d \cdot W_f \tag{5}$$

The proposed GCNN-DDGF model can further enhance the GRU gate operations with a data-driven graph filter and a spectral graph convolution. More details can be found in the following sub-sections.

3.2. Spectral Graph Convolution

The whole traffic network can be represented as a graph $G = (V, x, \varepsilon, A)$, where V is a finite set of vertices (i.e., traffic sensors) with size N; signal $x \in \mathbb{R}^N$ is a scalar for every vertex, e.g., the traffic volume for a specific hour; ε is a set of edges, $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix, and entry A_{ij} represents the connection degree between the signals at two vertices. A normalized graph Laplacian matrix is defined as

$$L = I_N - D^{-1/2} A D^{-1/2} \tag{6}$$

where I_N is the identity matrix, and $D \in \mathbb{R}^{N \times N}$ is a diagonal degree matrix with $D_{ii} = \sum_j A_{ij}$. L is a real symmetric positive semidefinite matrix that can be diagonalized as

$$L = U \Lambda U^T \tag{7}$$

where $U = [u_0, u_1, \ldots, u_{N-1}]$; $\Lambda = diag([_0, _1, \ldots, _{N-1}])$; $_0, _1, \ldots, _{N-1}$ are the eigenvalues of L, and $u_0, u_1, \ldots, u_{N-1}$ are the corresponding set of orthonormal eigenvectors.

A spectral convolution on the graph is defined as follows:

$$g_\theta \times x = U g_\theta(\Lambda) U^T x \tag{8}$$

where $g_\theta(\Lambda)$ is denoted as a function of the eigenvalues of L.

In Equation (8), the spectral graph convolution process includes graph Fourier transform, filtering, and inverse graph Fourier transform.

The time complexity defined in Equation (8) is $\mathrm{O}\left(N^2\right)$, due to the multiplication with the eigenvector matrix U. Moreover, the eigendecomposition of L is also computationally expensive. Hence, on the graph, a simplified spectral convolution is defined as follows [32]:

$$g_\theta \times x = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} x \theta \tag{9}$$

where $\theta \in \mathbb{R}$ is the filter parameter; $\widetilde{A} = A + I_N$ indicates the summation of the adjacency matrix of the undirected graph A and the identity matrix I_N. $\widetilde{A}$ is the adjacency matrix of an undirected graph where each vertex connects with itself; $\widetilde{D}_{ii} = \sum\limits_j \widetilde{A}_{ij}$. The computational complexity is reduced to $\mathrm{O}(|\varepsilon|)$.

3.3. Data-Driven Graph Filter

The spectral graph convolution usually needs a predefinition of adjacency matrix A; however, it is not trivial to determine. The graph structure is of critical importance to the GCNN performances [30,32]. In a previous study [10], Equation (9) could be redefined as the following:

$$g_{\hat{A}} \times x = \widetilde{D}^{-\frac{1}{2}} \hat{A} \widetilde{D}^{-\frac{1}{2}} x \tag{10}$$

where $\hat{A}$ is called the Data-driven Graph Filter (DDGF), which is a symmetric matrix consisting of trainable filter parameters, $\hat{A} \in \mathbb{R}^{N \times N}$.

Now we integrate the spectral graph convolution defined in Equation (10) into the gate operations in the GRU cell. Equations (1)–(3) would be straightforwardly revised as follows:

$$z_t = \sigma\left(g_{\hat{A}_z} \times [h_{t-1}, x_t] \cdot W_z\right) \tag{11}$$

$$r_t = \sigma\left(g_{\hat{A}_r} \times [h_{t-1}, x_t] \cdot W_r\right) \tag{12}$$

$$\widetilde{h}_t = tahn\left(g_{\hat{A}} \times [r_t \cdot h_{t-1}, x_t] \cdot W\right) \tag{13}$$

W_z, W_r, W, $\hat{A}_z$, $\hat{A}_r$, and $\hat{A}$ can be trained through the back-propagation procedure.

4. Data Preparation

An hourly traffic volume dataset from the PeMS system, which was collected from 150 sensors in the Los Angeles area from 1 January 2018 to 30 June 2019 [21], was used. The dataset is big enough, and each sensor recorded 13,104 hourly traffic volumes (vehicle per hour). The mean values of the traffic volume series from each sensor are ascendingly sorted and illustrated in Figure 3a, where the upper and lower bounds are also plotted based on the standard deviations. Figure 3b,c show 24 h of traffic volume data for a weekday (1 February 2018) from randomly selected two sensors. Sensor 716271 (Figure 3b) has two peak hours, and the largest one is around 1000 vehicles at 17:00, while the peak-hour traffic volume of Sensor 716500 (Figure 3c) in the morning has more than 10,000 vehicles. Some hourly traffic volumes for the afternoon are higher than 8000 vehicles.

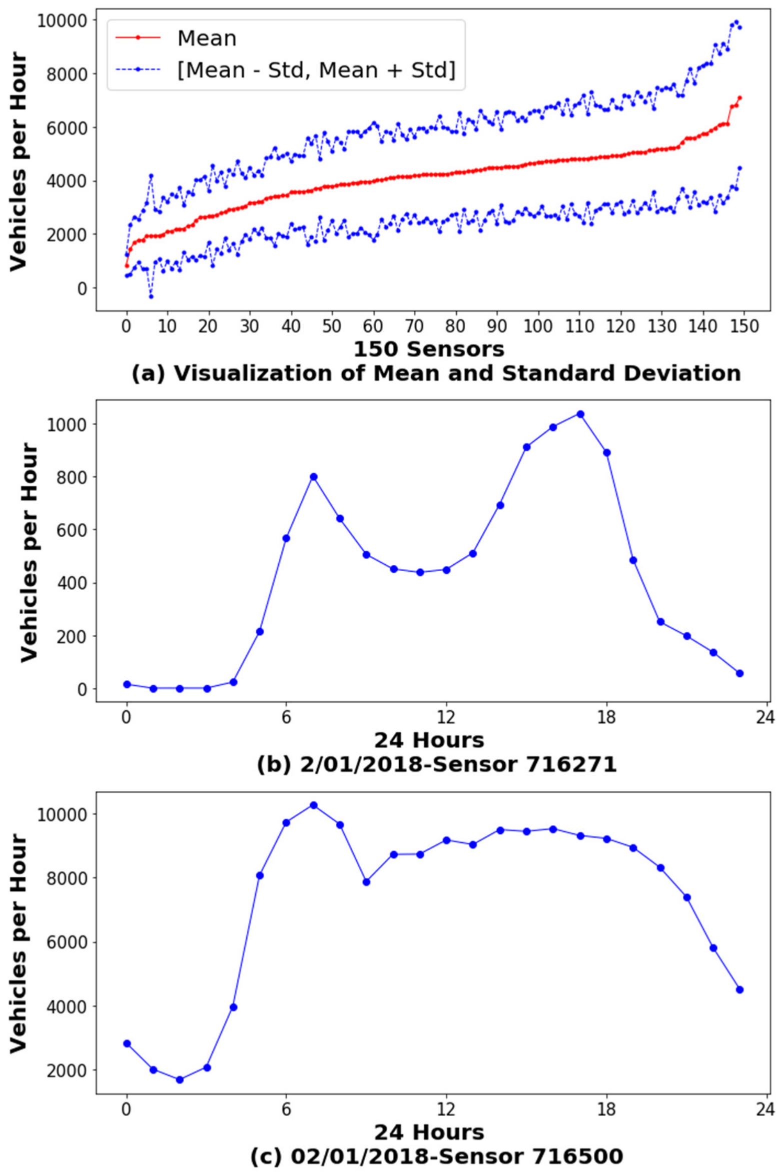

Figure 3. (**a**) Mean and Standard Deviation of Hourly Traffic Volumes for 150 Sensors; (**b**) Hourly Traffic Volumes on 1 February 2018 for Sensor 716271; (**c**) Hourly Traffic Volumes on 1 February 2018 for Sensor 716500.

Figure 4 further illustrates the locations of all 150 traffic sensors, which are distributed along with the highway system on the map. The sensors with the mean hourly traffic volumes in the top and bottom 25 percentiles are noted as red and black points. The blue points are the remaining sensors. According to the spatial distribution of the sensors, the west side of the network (more red sensors observed) may have a relatively higher traffic volume than the east side.

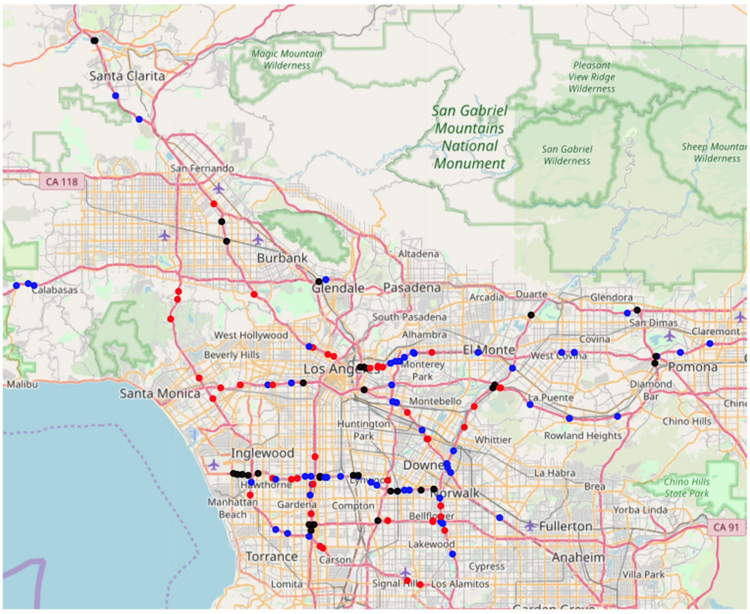

Figure 4. Locations of the 150 Sensors in Los Angeles Network.

5. Experimental Results

The whole dataset is split into training, validation, and testing datasets according to rates of 0.7, 0.1, and 0.2. The task of this study is to predict 12-step (hours) traffic volume in the future for all sensors by their historical 12-step traffic volume data. To verify the prediction performance of the developed model, a few benchmark models, including deep learning and statistical models, have been built into this study. A brief introduction of these benchmark models is listed below:

(1) **Diffusion Convolution Recurrent Neural Network (DCRNN)**: It outperforms multiple state-of-the-art machine learning models for network-wide multi-step traffic speed prediction [27]. DCRNN requires a predefined adjacency matrix according to the road network's topological features. In this study, the $(i,\ j)^{th}$ element was calculated as follows:

$$A_{ij} = \exp\left(-\frac{dist(v_i,\ v_j)}{\sigma^2}\right) \tag{14}$$

where $dist(v_i,\ v_j)$ is computed using the spatial distance between the sensor v_i and v_j; σ is the standard deviation of distances. Further, A_{ij} is set as 0 if $A_{ij} \leq 0.1$ to keep the sparsity of the adjacency matrix.

(2) **Sequence to Sequence Recurrent Neural Network (Seq2Seq-RNN) model**: Seq2Seq-RNN model has been mainly used in naturalistic language processing (NLP) studies [41]. The Seq2Seq-RNN model implemented in this study is designed for a high-dimension time series prediction [42].

(3) **Vector Autoregression (VAR) model**: Autoregression models (AR) have been widely used in traffic volume prediction studies [2,43]. VAR models taking high-dimension time series data as the input can predict future traffic volume for multiple sensors.

(4) **Linear Regression (LR) model**: A conventional linear regression model has been built for each station that takes traffic volume data from the previous 12 h to predict the next 12-h traffic volume.
(5) **Historical Average model (HA)**: HA is the simplest model that uses the mean of the traffic volume at the same hour across different days for the same sensor in the training dataset as the prediction.

5.1. Evaluation Metric

Three evaluation criteria are used for the comparison of these models, including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). The calculation equations are shown below:

$$MAE = \frac{1}{F*N}\sum_{i}^{F}\sum_{j}^{N}|y_{ij} - \hat{y}_{ij}| \tag{15}$$

$$MAPE = \frac{1}{F*N}\sum_{i}^{F}\sum_{j}^{N}\frac{|y_{ij} - \hat{y}_{ij}|}{y_{ij}} \tag{16}$$

$$RMSE = \sqrt{\frac{1}{F*N}\sum_{i}^{F}\sum_{j}^{N}(y_{ij} - \hat{y}_{ij})^2} \tag{17}$$

where $F = 12$ denotes the prediction horizon, while $N = 150$ indicates the sensor number. $\hat{y}_{ij}$ and y_{ij} are the predicted and ground truth traffic volume in hour i for sensor j, respectively.

5.2. Deep Learning Model Training

Training deep learning models, such as GCNN-DDGF, DCRNN, and Seq2Seq-RNN is not trivial. This study calibrates hyperparameters of deep learning models using the grid search method. An early-stopping mechanism is used to avoid overfitting. If the MAE of the validation dataset is reduced for 50 continuous epochs, the model training will continue. The testing dataset is used to evaluate the models with the lowest MAE tested in the validation dataset.

Figure 5 illustrates the training time for the three mentioned deep learning models. Seq2Seq-RNN only needs 32 min for training, which is much faster than GCNN-DDGF and DCRNN. This is because Seq2Seq-RNN does not have the graph convolution operation. The training of GCNN-DDGF is much quicker than the DCRNN due to the simplified spectral graph convolution mechanism.

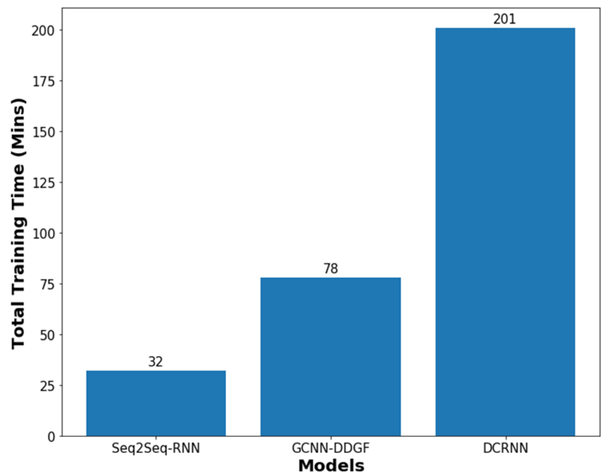

Figure 5. Training Time Comparison of Three Deep Learning Models.

5.3. Prediction Performance Comparison

As shown in Table 1, deep learning models (GCNN-DDGF, DCRNN, Seq2Seq-RNN) significantly perform better than the traditional models (VAR, LR, HA) in all performance metrics. Especially, GCNN-DDGF has the best overall performances for all three evaluation parameters due to its trainable adjacency matrices. It is followed by the other two deep learning models, DCRNN and Seq2Seq-RNN. Compared to the Seq2Seq-RNN model, the graph convolution operations of GCNN-DDGF and DCRNN contribute to higher accuracy. For the traditional models, the VAR model performs better than LR and HA.

Table 1. Overall Performances on Testing Dataset.

	MAE	MAPE	RMSE
GCNN-DDGF	**343.37**	**14.19%**	**540.69**
DCRNN	459.85	17.84%	763.76
Seq2Seq-RNN	521.01	21.12%	821.51
VAR	640.64	27.03%	895.31
LR	788.13	33.37%	1087.38
HA	752.98	35.36%	1086.81

5.4. Performance Comparison from Temporal Aspect

Considering the main task of the study is to predict the future multiple-step traffic volume, it is interesting to compare the MAE, MAPE, and RMSE of each prediction step for all six models. As shown in Figure 6, as the prediction step goes up, prediction errors of all models except the HA model generally increase, although a slight prediction error drop is observed for some models. The constant prediction error of the HA model can be explained by the fact that the HA model does not rely on the 12 historic steps to make predictions but simply uses the average traffic volume for the same hour and sensor. As expected, the deep learning models outperform all the prediction steps in terms of prediction accuracy. The proposed GCNN-DDGF keeps the best performance, except that the DCRNN model has the lowest MAPE for the first prediction step.

Forecasting at a far distance is usually hard and unreliable. The error curves of DCRNN and Seq2Seq-RNN slightly go down at later prediction steps. The proposed model shows a similar pattern but with better stability. Interestingly, the error of multi-step prediction does not necessarily increase monotonically. Previous research has also found this phenomenon without further discussion [44,45]. This may be because the prediction accuracy at a particular step depends on many factors, including the number of steps, the quality of previous predictions, and the time of the day. Predicting traffic during peak hours is more complicated than off-peak hours [46]. As we conduct 12-step prediction at one hour per step, the prediction time may range both “easy-to-predict” and “difficult-to-predict” hours, thus affecting the error curve shape.

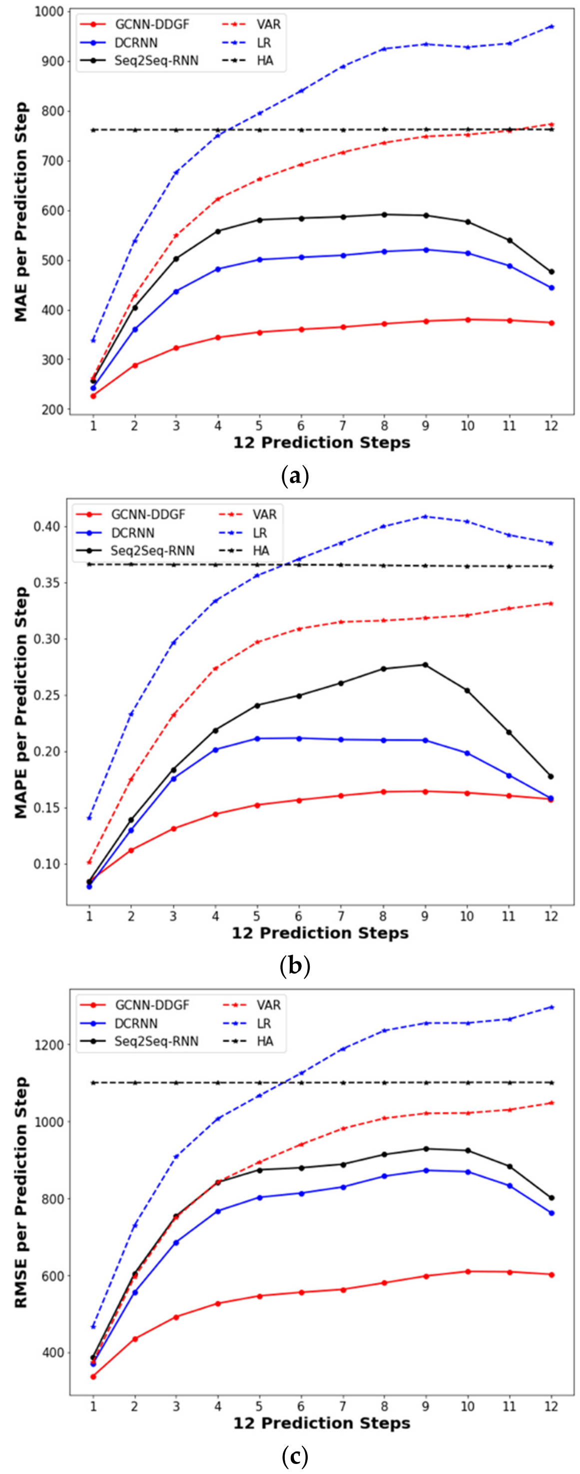

Figure 6. Model Comparison for each Prediction Step. (**a**) MAE; (**b**) MAPE; (**c**) RMSE.

6. Performance Comparison from Spatial Aspect

Figure 7 shows the histograms of MAE, MAPE, and RMSE for each sensor for GCNN-DDGF and DCRNN models. In general, the proposed GCNN-DDGF model has better prediction accuracy. For example, Figure 7a shows that from GCNN-DDGF, most sensors have MAEs between 200 and 400, while Figure 7d shows that from DCRNN, the range between 400 and 600 has a higher frequency. Similarly, GCNN-DDGF generates more sensors with lower MAPEs and RMSEs than DCRNN from Figure 7.

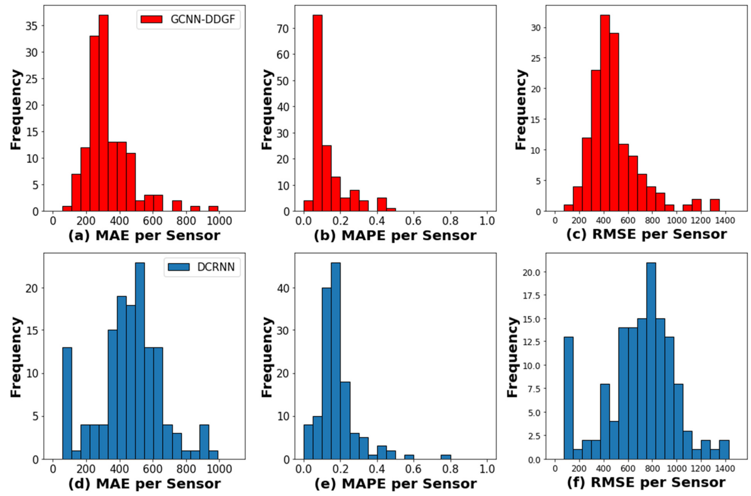

Figure 7. Model Comparison for each Sensor. (**a**–**c**) MAE, MAPE, and RMSE for GCNN-DDGF; (**d**–**f**) MAE, MAPE, and RMSE for DCRNN.

Figure 8 takes a closer look at the prediction made by GCNN-DDGF. We randomly pick three sensors: 716271, 716067, and 716032. For each sensor, two 12-step predictions, one at 00:00 and the other one at 12:00, are displayed. The corresponding ground truth values are also plotted. Sensor 716271 has a much lower traffic volume compared with the other two. The MAE for one 12-step prediction ranges from 67.48 to 502.88. In general, the predicted increasing or decreasing trends are consistent with the real data trends, and the predictions are close to the actual values for the nighttime. In contrast, a few hours in the daytime have more significant deviations, such as 8:00, 15:00, 18:00, and so on.

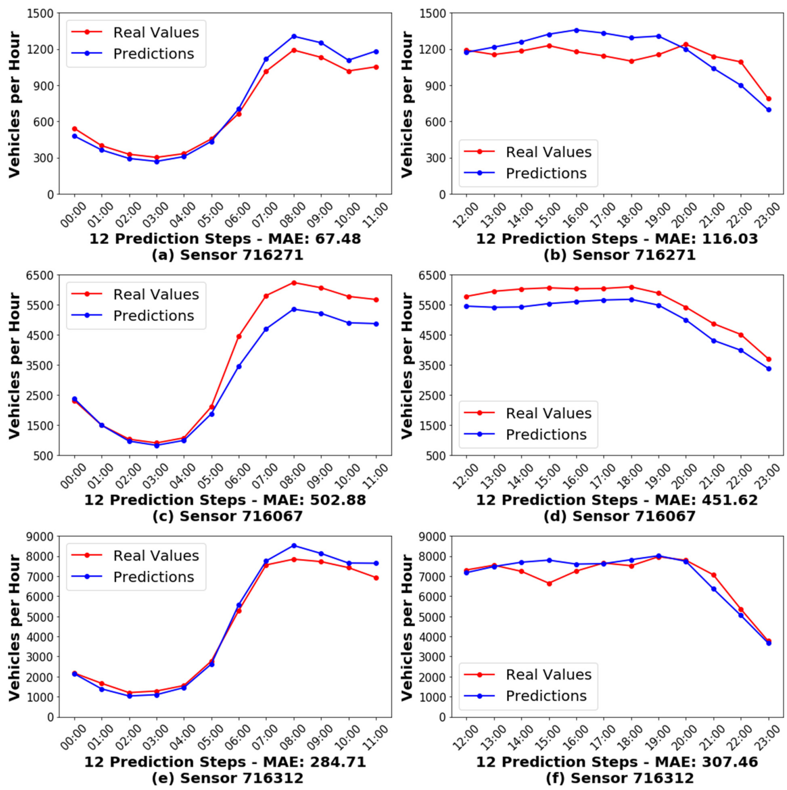

Figure 8. Visualization of Predictions from GCNN-DDGF. (**a**,**c**,**e**) are 12-step predictions made at 12:00 a.m. (midnight) for sensors 716271, 716067, and 716312. (**b**,**d**,**f**) are predictions made at 12:00 p.m. for the same three sensors.

7. Conclusions and Future Research

This study proposes a graph convolutional neural network with a data-driven graph filter (GCNN-DDGF) model for network-wide traffic volume prediction, which does not require the predefinition of the adjacency matrix for graph convolution; instead, it integrates the DDGF with gated recurrent network (i.e., GRU) cell which can automatically capture the hidden spatiotemporal correlation between state sensors. The sequence-to-sequence architecture also allows it to utilize the temporal dependency for multi-step-ahead predictions. The GCNN-DDGF model was tested on a real-world dataset along with two deep learning models and three classical benchmark models. The proposed model consistently outperforms the others in terms of MAE, MAPE, and RMSE. For instance, the GCNN-DDGF model improves MAE, MAPE, and RMSE by 25.33%, 20.45%, and 29.20%, compared to other deep learning models, such as Diffusion Convolution Recurrent Neural Network (DCRNN). Furthermore, the GCNN-DDGF has also demonstrated better performance for both temporal and spatial aspects.

For future research directions, it is interesting to analyze the learned adjacency matrices further to understand the correlations between traffic sensors. More tests for applying the proposed model to different traffic state parameters, such as traffic speed, are necessary if other traffic state datasets are available. The unobserved heterogeneity may be considered for developing a more sophisticated model in the future.

Author Contributions: Conceptualization, L.L. and L.Z.; methodology, L.L. and L.Z.; software, L.L.; validation, L.L., W.L. and L.Z.; formal analysis, L.L. and W.L.; data curation, L.L.; writing—original draft preparation, L.L. and L.Z.; writing—review and editing, W.L.; visualization, L.L. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Chandra, S.R.; Al-Deek, H. Predictions of Freeway Traffic Speeds and Volumes Using Vector Autoregressive Models. *J. Intell. Transp. Syst.* **2009**, *13*, 53–72. [CrossRef]
2. Lin, L.; Wang, Q.; Sadek, A. Short-Term Forecasting of Traffic Volume: Evaluating Models Based on Multiple Data Sets and Data Diagnosis Measures. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2392*, 40–47. [CrossRef]
3. Anacleto, O.; Queen, C.; Albers, C.J. Multivariate forecasting of road traffic flows in the presence of heteroscedasticity and measurement errors. *J. R. Stat. Soc. Ser. C* **2013**, *62*, 251–270. [CrossRef]
4. Lin, L.; Li, Y.; Sadek, A. A k Nearest Neighbor based Local Linear Wavelet Neural Network Model for On-line Short-term Traffic Volume Prediction. *Procedia-Soc. Behav. Sci.* **2013**, *96*, 2066–2077. [CrossRef]
5. Zhang, Y.; Liu, Y. Traffic forecasting using least squares support vector machines. *Transportmetrica* **2009**, *5*, 193–213. [CrossRef]
6. Lin, L.; Handley, J.C.; Gu, Y.; Zhu, L.; Wen, X.; Sadek, A.W. Quantifying uncertainty in short-term traffic prediction and its application to optimal staffing plan development. *Transp. Res. Part C Emerg. Technol.* **2018**, *92*, 323–348. [CrossRef]
7. Hua, J.; Faghri, A. Applications of Artificial Neural Networks to Intelligent Vehicle-Highway Systems. *Transp. Res. Rec.* **1994**, *1453*, 83.
8. Le Callet, P.; Viard-Gaudin, C.; Barba, D. A Convolutional Neural Network Approach for Objective Video Quality Assessment. *IEEE Trans. Neural Netw.* **2006**, *17*, 1316–1327. [CrossRef]
9. Shen, Y.; He, X.; Gao, J.; Deng, L.; Mesnil, G. Learning semantic representations using convolutional neural networks for web search. In Proceedings of the 23rd International Conference on World Wide Web—WWW'14 Companion, Seoul, Republic of Korea, 7–11 April 2014; ACM Press: New York, NY, USA; pp. 373–374. [CrossRef]
10. Lin, L.; He, Z.; Peeta, S. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transp. Res. Part C Emerg. Technol.* **2018**, *97*, 258–276. [CrossRef]
11. Ma, X.; Tao, Z.; Wang, Y.; Yu, H.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. Part C Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]

12. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.-Y. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [CrossRef]
13. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17. [CrossRef]
14. Wu, Y.; Tan, H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv* **2016**, arXiv:1612.01022.
15. 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. Part C Emerg. Technol.* **2016**, *62*, 21–34. [CrossRef]
16. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
17. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2 (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; MIT Press: Cambridge, MA, USA, 2014; pp. 3104–3112.
18. Ma, X.; Dai, Z.; He, Z.; Na, J.; Wang, Y.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *arXiv* **2017**, arXiv:1701.04245. [CrossRef]
19. Jia, Y.; Wu, J.; Xu, M. Traffic Flow Prediction with Rainfall Impact Using a Deep Learning Method. *J. Adv. Transp.* **2017**, *2017*, 1–10. [CrossRef]
20. Lin, L.; Ni, M.; He, Q.; Gao, J.; Sadek, A.W. Modeling the Impacts of Inclement Weather on Freeway Traffic Speed: Exploratory Study with Social Media Data. *Transp. Res. Rec. J. Transp. Res. Board* **2015**, *2482*, 82–89. [CrossRef]
21. Caltrans Caltrans PeMS. Available online: http://pems.dot.ca.gov/ (accessed on 27 January 2020).
22. Zhu, L.; Gonder, J.; Lin, L. Prediction of Individual Social-Demographic Role Based on Travel Behavior Variability Using Long-Term GPS Data. *J. Adv. Transp.* **2017**, *2017*, 7290248. Available online: https://www.hindawi.com/journals/jat/2017/7290248/ (accessed on 22 May 2018). [CrossRef]
23. Zhu, L.; Holden, J.R.; Gonder, J.D. Trajectory Segmentation Map-Matching Approach for Large-Scale, High-Resolution GPS Data. *Transp. Res. Rec. J. Transp. Res. Board* **2017**, *2645*, 67–75. [CrossRef]
24. Lin, L.; Li, W.; Peeta, S. Efficient Data Collection and Accurate Travel Time Estimation in a Connected Vehicle Environment via Real-Time Compressive Sensing. *J. Big Data Anal. Transp.* **2019**, *1*, 95–107. [CrossRef]
25. Lin, L.; Pccta, S.; Wang, J. Efficient Collection of Connected Vehicle Data based on Compressive Sensing*. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3427–3432. [CrossRef]
26. Zhang, J.; Zheng, Y.; Qi, D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
27. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv* **2017**, arXiv:1707.01926.
28. Min, W.; Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 606–616. [CrossRef]
29. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *arXiv* **2017**, arXiv:1709.04875.
30. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29*; Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 3844–3852.
31. Sandryhaila, A.; Moura, J.M.F. Discrete Signal Processing on Graphs. *IEEE Trans. Signal Process.* **2013**, *61*, 1644–1656. [CrossRef]
32. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
33. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
34. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Spatio-Temporal Short-Term Urban Traffic Volume Forecasting Using Genetically Optimized Modular Networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2007**, *22*, 317–325. [CrossRef]
35. Zou, Y.; Hua, X.; Zhang, Y.; Wang, Y. Hybrid short-term freeway speed prediction methods based on periodic analysis. *Can. J. Civ. Eng.* **2015**, *42*, 570–582. [CrossRef]
36. Li, L.; Qin, L.; Qu, X.; Zhang, J.; Wang, Y.; Ran, B. Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. *Knowl.-Based Syst.* **2019**, *172*, 1–14. [CrossRef]
37. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Boston, MA, USA, 2016.
38. Wang, C.; Hou, Y.; Barth, M. Data-Driven Multi-step Demand Prediction for Ride-Hailing Services Using Convolutional Neural Network. In *Advances in Computer Vision*; Arai, K., Kapoor, S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 11–22. [CrossRef]
39. Yu, B.; Song, X.; Guan, F.; Yang, Z.; Yao, B. k-Nearest Neighbor Model for Multiple-Time-Step Prediction of Short-Term Traffic Condition. *J. Transp. Eng.* **2016**, *142*, 04016018. [CrossRef]
40. Shao, L.; Gouws, S.; Britz, D.; Goldie, A.; Strope, B.; Kurzweil, R. Generating High-Quality and Informative Conversation Responses with Sequence-to-Sequence Models. *arXiv* **2017**, arXiv:1701.03185.

41. Serban, I.V.; Klinger, T.; Tesauro, G.; Talamadupula, K.; Zhou, B.; Bengio, Y.; Courville, A. Multiresolution Recurrent Neural Networks: An Application to Dialogue Response Generation. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
42. Chevalier, G. Guillaume-Chevalier/seq2seq-Signal-Prediction. 2019. Available online: https://github.com/guillaume-chevalier/seq2seq-signal-prediction (accessed on 27 October 2022).
43. Lin, L.; Wang, Q.; Huang, S.; Sadek, A.W. On-line prediction of border crossing traffic using an enhanced Spinning Network method. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 158–173. [CrossRef]
44. Hao, S.; Lee, D.-H.; Zhao, D. Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system. *Transp. Res. Part C Emerg. Technol.* **2019**, *107*, 287–300. [CrossRef]
45. Muralidhar, N.; Mu-thiah, S.; Nakayama, K.; Sharma, R.; Ramakrishnan, N. Multivariate long-term state forecasting in cyber-physical systems: A sequence to sequence approach. In Proceedings of the 2019 IEEE In-ternational Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 543–552.
46. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume. *Transp. Res. Part C Emerg. Technol.* **2006**, *14*, 351–367. [CrossRef]