electronics

MDPI

Article

LST-GCN: Long Short-Term Memory Embedded Graph Convolution Network for Traffic Flow Forecasting

Xu Han and Shicai Gong *

College of Science, Zhejiang University of Science and Technology, Hangzhou 310023, China;
222009252009@zust.edu.cn
* Correspondence: scgong@zafu.edu.cn; Tel.: +86-137-7758-5486

Abstract: Traffic flow prediction is an important part of the intelligent transportation system. Accurate traffic flow prediction is of great significance for strengthening urban management and facilitating people's travel. In this paper, we propose a model named LST-GCN to improve the accuracy of current traffic flow predictions. We simulate the spatiotemporal correlations present in traffic flow prediction by optimizing GCN (graph convolutional network) parameters using an LSTM (long short-term memory) network. Specifically, we capture spatial correlations by learning topology through GCN networks and temporal correlations by embedding LSTM networks into the training process of GCN networks. This method improves the traditional method of combining the recurrent neural network and graph neural network in the original spatiotemporal traffic flow prediction, so it can better capture the spatiotemporal features existing in the traffic flow. Extensive experiments conducted on the PEMS dataset illustrate the effectiveness and outperformance of our method compared with other state-of-the-art methods.

Keywords: traffic flow forecasting; long short-term memory network; graph convolutional network

check for updates

Citation: Han, X.; Gong, S. LST-GCN: Long Short-Term Memory Embedded Graph Convolution Network for Traffic Flow Forecasting. *Electronics* **2022**, *11*, 2230. https://doi.org/10.3390/electronics11142230

Academic Editor: Wojciech Mazurczyk

Received: 16 June 2022
Accepted: 16 July 2022
Published: 17 July 2022

1. Introduction

In recent years, with the increase in the utilization rate of automobiles, the traffic flow on the road is increasing day by day. When the road is insufficient to accommodate vehicles, problems such as traffic congestion and traffic accidents will emerge. In this situation, traffic flow prediction is of great significance [1,2]. Traffic flow prediction refers to an analysis using traffic flow, speed and other information obtained by sensors in a certain road section for future prediction. It provides effective assistance in planning driving routes, thereby avoiding potential traffic jams.

Traffic flow prediction is inseparable from the temporal and spatial information in the road network. Individually considering any aspect of the information in the prediction will lead to a lack of information, and hence affect the accuracy of prediction. We need to predict outcomes from both a temporal and spatial perspective. Traffic data are recorded at fixed time points and fixed locations in space. Observations at adjacent locations and adjacent timestamps are not independent of each other, but are dynamically related. The key to such tasks is to explore dynamic correlations in data space and time to make accurate predictions.

With the advancement of technology, it has become easier to obtain data about the transportation networks, which also makes it more convenient for us to predict the traffic flow. Using cameras, sensors and other equipment on the highway, people can collect a large amount of time-series data, including traffic flow, speed, occupancy, and other information, which provides a solid data foundation for traffic forecasting, thus giving birth to a series of traffic forecast methods [3]. These include statistical methods and machine-learning methods. These methods either rely on feature engineering or cannot consider both the time and space information of the data and have certain limitations in the prediction of traffic flow. With the development of deep learning, some researchers tried to use graph convolutional networks to predict traffic flow or combine graph convolutional networks with recurrent neural networks to capture spatial and temporal features in traffic flow. Although much progress has been made in the prediction of traffic flow, most studies do not consider the periodicity of traffic flow, so the prediction of traffic flow still does not achieve the desired accuracy. To improve the accuracy of model predictions, we take into account the weekly and daily periodicity of traffic flow.

To make a more accurate traffic flow prediction, the LST-GCN model is proposed in this paper, and the LSTM model [4] is embedded into the parameter training of the GCN model [5], to capture the time and space information more synchronously. Further, we explore the internal relation of time and space, and reduce the number of parameter training, so as to make more accurate prediction.

The original combined model is relatively simple in processing data sets, such as the combined model of LSTM model and GCN model. For traffic flow data, the GCN model is used to update the node flow information at each moment separately to obtain data space information, and then using the LSTM model further combines the node traffic information at all times to obtain information about the time of the data. The disadvantage of this method is that the number of model parameters and calculations are large. In response to this problem, we propose a new LST-GCN embedded structure. Different from previous models, we directly embed the LSTM model into the update process of GCN parameters, which greatly reduces the number of parameters and the amount of computation. At the same time, the model can make good use of the temporal and spatial information of the data.

The remainder of this paper is organized as follows. The related works on traffic flow forecasting are discussed in Section 2. In Section 3, we propose some definitions about traffic flow and introduce the structure of the GCN model and LSTM models. Section 4 proposes the LST-GCN model to capture spatial correlations by learning topology through GCN networks and temporal correlations by embedding LSTM networks into the training process of GCN networks. In Section 5, a comprehensive assessment of the model performance is conducted using real road-traffic datasets. At the same time, the experimental results are discussed. Section 6 concludes the paper and provides an outlook on future work.

2. Related Work

2.1. Traffic Forecasting

There are two main types of methods for traffic flow forecasting: one is the statistical method and the other is the machine-learning method. The statistical methods mainly include ARIMA (autoregressive integrated moving average model) [6–8], HA (history average model) [3], ES (exponential smoothing model) [9] and KF (Kalman filter model) [10–13]. ARIMA models analyze time-series data and use them to make predictions about future traffic flows. The ARIMA model [6–8] assumes that the change in traffic flow is linear. The HA model [2] uses the least-squares method to evaluate the parameters of the model to further predict the traffic flow. The ES model [9] and the KF model [10–13] are suitable for making predictions on traffic flow with a smaller amount of data. The assumptions of these models are relatively strict. Once random interference occurs, the accuracy of the models will decrease. They rely on the assumption of stability. At the same time, these models cannot reflect the nonlinearity of traffic conditions. Therefore, the use of these models has certain limitations.

There are many machine-learning methods for traffic flow prediction, which are mainly divided into two categories: the traditional machine-learning method and the deep-learning method. The SVR (support vector regression) model [14], KNN (K-nearest neighbor) model [15], Bayesian model [16], fuzzy logic model [17], neural-network model [18], etc., as traditional machine-learning methods, are often used to predict traffic flow. The SVR model [14] introduces a supervised machine-learning method called regressive online support vector machines, which can make short-term traffic flow predictions for both typical and atypical conditions. The KNN model [15] takes the k value and d_m value of the nearest neighbors as the input parameters of the model, and combines the prediction range of multiple intervals to optimize the parameter values of the model, and then predict the value of traffic flow. The Bayesian model [16] first searches the manifold neighborhood, and then obtains a higher accuracy of the manifold neighborhood, and then proposes a traffic-state prediction method based on the expansion strategy of adaptive neighborhood selection. Fuzzy logic models [17] use fuzzy methods to classify input data into clusters, which in turn specify input–output relationships. The neural-network model [18] is the first attempt to build an artificial neural network based on historical traffic data, aiming to predict traffic volume based on historical data at major urban intersections. This type of model has strong nonlinear mapping ability, and the data requirements are not as strict as statistical methods, so it can better adapt to the uncertainty of traffic flow and effectively improve the prediction effect. However, the spatial structure of observation points is unstructured, and the above methods do not use the spatial structure information of the data, and only analyzing from the time dimension has certain limitations in improving the prediction accuracy.

The deep-learning models originally used for traffic flow prediction mainly include the GRU (gated recurrent unit) model [19] and LSTM model. The GRU model and LSTM model are important recursive neural-network models that are used to integrate and analyze temporal information to make predictions. Compared with the prediction models based on statistical learning and machine-learning methods, deep learning can model multidimensional features and realize the approximation of complex functions by learning the deep nonlinear network structures, which can better learn the abundant changes inherent in traffic flow. It can simulate its complex nonlinear relationship and greatly improve the accuracy of traffic flow prediction. However, these models also did not consider the influence of the spatial structure of the data on the prediction results, and did not fully mine the spatiotemporal characteristics of the traffic data. There are also certain limitations in predicting traffic flow.

Recently, models that consider spatiotemporal information have sparked a lot of research. Wu et al. [20] designed a feature fusion framework for short-term traffic flow prediction by combining the CNN (convolutional neural network) model with the LSTM model. This framework uses a one-dimensional CNN to describe the spatial features of traffic flow data. For the time-varying periodicity and temporal variation of the traffic flow, this framework utilizes two LSTM models. DCRNN, proposed by Li et al. [21], uses a bidirectional random walk to capture spatial dependencies and an encoder-decoder with predetermined sampling to capture temporal dependencies. Sun et al. [22] constructed a multibranch framework called TFPNet (traffic flow prediction network), a deep-learning framework for short-term traffic flow prediction. TPFNet uses a multilayer fully convolutional network structure to extract the relationship from local to global hierarchical space. Zhao et al. [23] proposed the T-GCN model, which combines gated recurrent units with graph convolutional networks for short-term traffic flow prediction. Geng et al. [24] designed a spatiotemporal multigraph convolutional network that first encodes the non-Euclidean pairwise correlations between regions into multiple graphs, and then uses multigraph convolution to explicitly map these correlations. Diao et al. [25] used a dynamic Laplacian matrix estimator to discover changes in the Laplacian matrix, which in turn made predictions about traffic flow. Huang et al. [26] proposed the cosAtt model, a graph-attention network that integrates cosAtt and GCN into a spatial gating block. Lv et al. [27] modeled various global features in road networks, including spatial, temporal, and semantic correlations, and proposed a temporal multigraph convolutional network. Guo et al. [28] used the attention mechanism for traffic flow prediction and proposed an AST-GCN model. The attention mechanism has been applied in both time and space and achieved better prediction results.

2.2. Convolutions on Graphs

In order to solve the irregularity of the spatial neighborhood, Bruna et al. [29] made a breakthrough from the spectral space and proposed a spectral network on the graph. According to the knowledge of graph theory, they decompose the Laplacian matrix spectrally and use the obtained eigenvalues and eigenvectors to define the convolution operation in the spectral space. To simplify the problem of complexity, Defferrard et al. [30] proposed a Chebyshev network, which defined the convolution kernel as a polynomial form, and used Chebyshev expansion to approximate the calculation of the convolution kernel, which greatly improved the computational efficiency. After that, Kipf and Welling [5] simplified the Chebyshev network, using only a first-order approximate convolution kernel, and made a little sign change, resulting in the well-known graph-convolution network.

2.3. Long Short-Term Memory Network

Bengio et al. [31] proposed the RNN (recurrent neural network) model. Using the RNN model can help people process sequence data more efficiently. In the RNN model, people can reinput the output of a neuron at a certain time as the input to the neuron. For the dependencies between time-series data, the network structure of the RNN model can adequately maintain them. However, this model suffers from vanishing gradients and exploding gradients. To solve the problems of gradient disappearance and gradient explosion in the traditional RNN model, Hochreiter et al. [4] proposed the LSTM network. The LSTM network is improved from the traditional RNN model. Compared with the RNN model, the hidden unit of the LSTM model has more complexity. At the same time, the LSTM model has a wider range of applications than RNN and is a more effective sequence model. During the run of the model, the LSTM model can selectively add or subtract information by adding linear interventions.

3. Preliminaries

3.1. Traffic Networks

Definition 1. *Road network G. We use $G = (V, E, A)$ to denote a spatial network, as shown in Figure 1, where $|V| = N$ is the set of vertices and N is the number of vertices. E is the set of edges, which reflects the connections between road sections. $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of the network G. The value of each element represents the connectivity between the corresponding road segments. An element value of 1 indicates connectivity, and an element value of 0 indicates disconnection.*

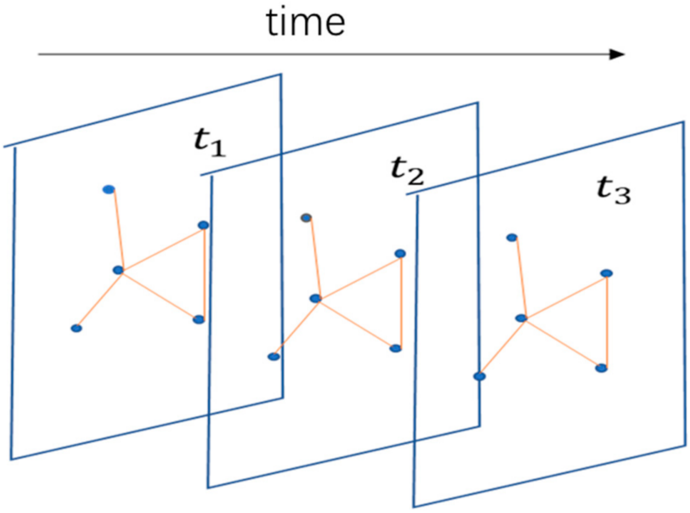


Figure 1. The spatial-temporal structure of traffic data, where the data at each time slice form a graph.

Definition 2. *The graph feature matrix $X_{\mathcal{G}}^{(t)} \in \mathbb{R}^{N \times C}$, where C is the number of attribute features and t represents the time step. The graph signal matrix represents the observations of the spatial network G at the time step t.*

The problem of traffic flow data prediction can be described as learning a mapping function, f, which maps the historical spatiotemporal network sequence $\left(X_{\mathcal{G}}^{(t-T+1)}, X_{\mathcal{G}}^{(t-T+2)}, \ldots, X_{\mathcal{G}}^{(t)}\right)$ into future observations of this spatiotemporal network $\left(X_{\mathcal{G}}^{(t+1)}, X_{\mathcal{G}}^{(t+2)}, \ldots, X_{\mathcal{G}}^{(t+T')}\right)$, where T represents the length of the historical spatiotemporal network sequence and T' denotes the length of the target spatiotemporal network sequence to be predicted.

3.2. GCN Model

Based on the Chebyshev network, Kipf and Welling proposed the GCN model. The updated convolution formula of each layer of the GCN model node is as follows:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right), \tag{1}$$

$$\widetilde{A} = A + I_N \tag{2}$$

$$\text{and } \widetilde{D} = D + I_N \tag{3}$$

Among them, $H^{(l+1)}$ represents the node representation of the $l+1$-th layer, $H^{(l)}$ represents the node representation of the $l+1$-th layer, and $W^{(l)}$ represents the learnable parameters of the l-th layer. A represents the adjacency matrix, I_N represents the identity matrix, and D represents the degree matrix.

By determining the topological relationship between the central node and the surrounding nodes, the GCN model can simultaneously encode the topological structure of the road network and the attributes of the nodes, so that spatial dependencies can be captured on this basis.

3.3. LSTM Model

The LSTM model is a typical RNN (recurrent neural network) model, which is proposed to solve the problems of gradient disappearance and gradient explosion existing in the traditional RNN model. The structure diagram of LSTM is shown in Figure 2, and the Equations are shown in (4)~(9).

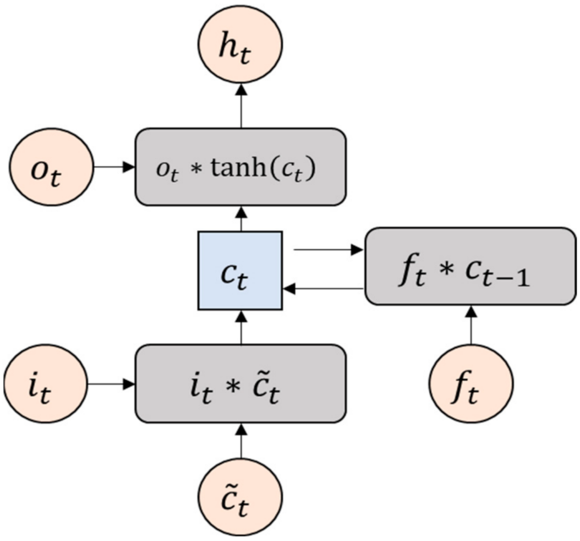


Figure 2. LSTM model diagram.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{4}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right), \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{6}$$

$$\widetilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c), \tag{7}$$

$$c_t = i_t * \widetilde{c}_t + f_t * c_{t-1} \tag{8}$$

$$\text{and } h_t = o_t * \tanh(c_t). \tag{9}$$

where i_t controls the input of the input gate to $\widetilde{c}_t$, f_t controls the memory level of the forget gate for c_{t-1}, and o_t controls the output of $\tanh(c_t)$. Since the activation function is a sigmoid function, the values of i_t, f_t, and o_t are in between 0 and 1.

The LSTM model uses the hidden state of the previous moment and the parameter information of the current moment as input to determine the parameter state of the current moment. Due to the gating mechanism, the LSTM model retains the changing trend of historical parameter information when capturing the parameter information at the current moment. Therefore, the model can capture the time-varying features of traffic dynamics from parametric data. In this paper, we apply the LSTM model to learn the temporal-varying trend of traffic states.

4. Method

Figure 3 shows the general framework of the LST-GCN model. The model consists of three parts with the same structure, and the model is established by representing data from three perspectives: adjacent time, daily cycle, and weekly cycle. As shown in Figure 3, this paper takes χ_h, χ_d, and χ_w as input, respectively. We consider each sensor as a node, and the sensor information about the three dimensions of traffic flow, vehicle speed, and occupancy rate is regarded as the vector representation of the node. χ_h, χ_d, and χ_w represent the node representation of all nodes at the adjacent time, the daily cycle, and the weekly cycle, respectively.

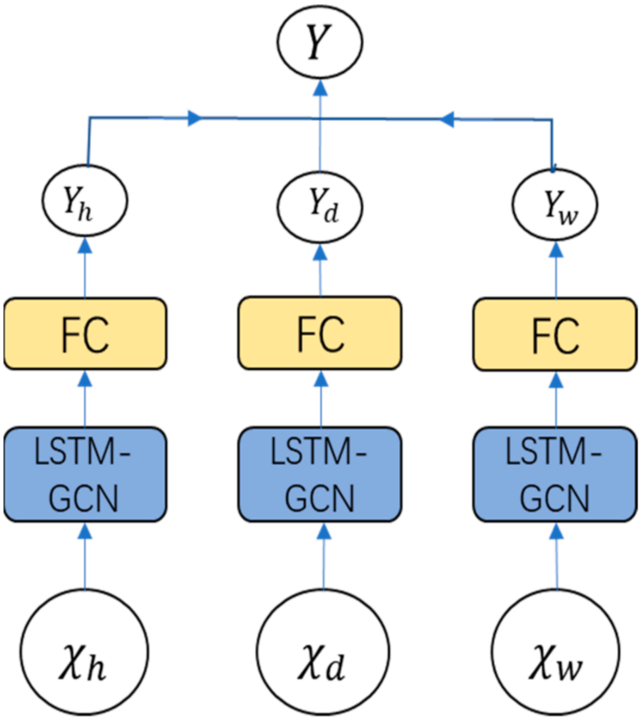


Figure 3. LST-GCN model frame diagram.

$\mathcal{X}_h \in \mathbb{R}^{N\times F\times T}$, N represents the number of nodes; the value of F is 3, which represents the three dimensions of traffic flow, vehicle speed, and occupancy; and T represents the length of the adjacent time slice.

We update the node representation through the LSTM-GCN block, and then use a fully connected layer to make predictions, and the results are denoted by Y_h, Y_d, and Y_w, respectively. Afterwards, the prediction results of the three series of proximity correlation, daily correlation, and weekly correlation are weighted and combined to obtain the final result, which is represented by Y.

Figure 4 shows the general framework of the LSTM-GCN block. Taking χ_h as an example, we take $X_{t_0-h+1}, X_{t_0-h+2}, \ldots, X_{t_0}$ as input. $\mathcal{X}_h \in \mathbb{R}^{N\times F\times T}$, N represents the number of nodes; the value of F is 3, which represents the three dimensions of traffic flow, vehicle speed, and occupancy; and T represents the length of the adjacent time slice. $X_{t_0-h+1}, X_{t_0-h+2}, \ldots, X_{t_0}$ represents the representation of each moment of χ_h. Through the LSTM-GCN block, we can update the node representation to obtain $X^1_{t_0-h+1}$, $X^1_{t_0-h+2}, \ldots, X^1_{t_0}$. Through the connection between the parameters, all GCN models are combined together and the representation of each vector is updated in time and space.

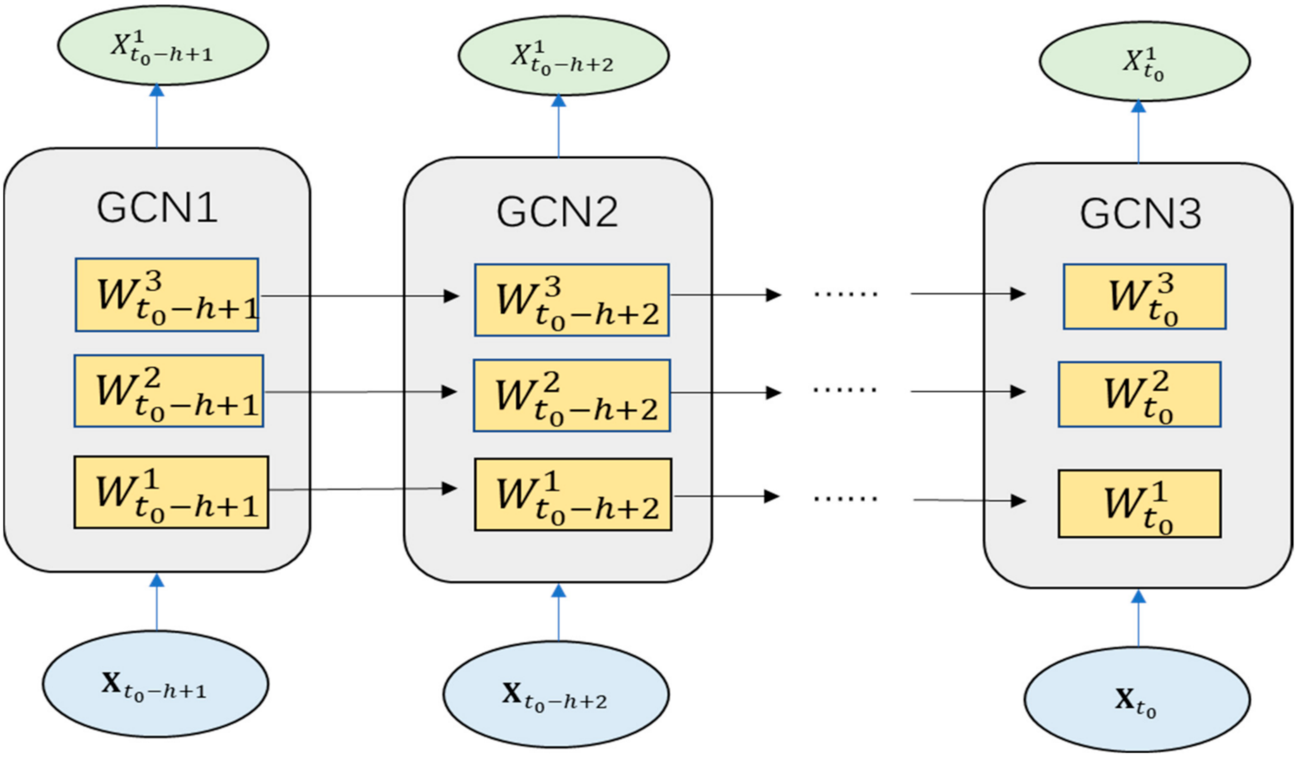


Figure 4. LSTM-GCN block diagram.

To explore the distribution of data from the perspective of space and time simultaneously, we introduce the LSTM model into the parameter update process of the GCN model. For the parameter $W^{(l)}$, we connect the $W^{(l)}$ at each moment through the LSTM model, as shown in Equation (10).

$$W_t^{(l)} = \mathrm{LSTM}\left(W_{t-1}^{(l)}\right) \tag{10}$$

Meanwhile, at time t, the convolution operation from the lth layer to the $l+1$-th layer is the same as that of the GCN model, as shown in Equation (11).

$$H_t^{(l+1)} = \mathrm{GCONV}\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}, H_t^{(l)}, W_t^{(l)}\right) \tag{11}$$

Combining Equations (10) and (11), we can obtain the update rule of node representation at $l + 1$-th layer, as shown in Equation (12).

$$\left[H_t^{(l+1)}, W_t^{(l)}\right] = \mathrm{LST} - \mathrm{GCN}\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}, H_t^{(l)}, W_{t-1}^{(l)}\right) \tag{12}$$

Figure 5 illustrates the update of the node. At time t, the representation of the node at $l + 1$-th layer is determined by the node and the parameters at l-th layer through convolution. Similarly, we can calculate the node representation of any layer. The node at the zeroth layer at time t is represented X_t corresponding to time t, that is, the vector representation of each sensor in the three dimensions of traffic flow, vehicle speed, and occupancy at time t. For the parameter W of each layer, we can update it through the LSTM model.

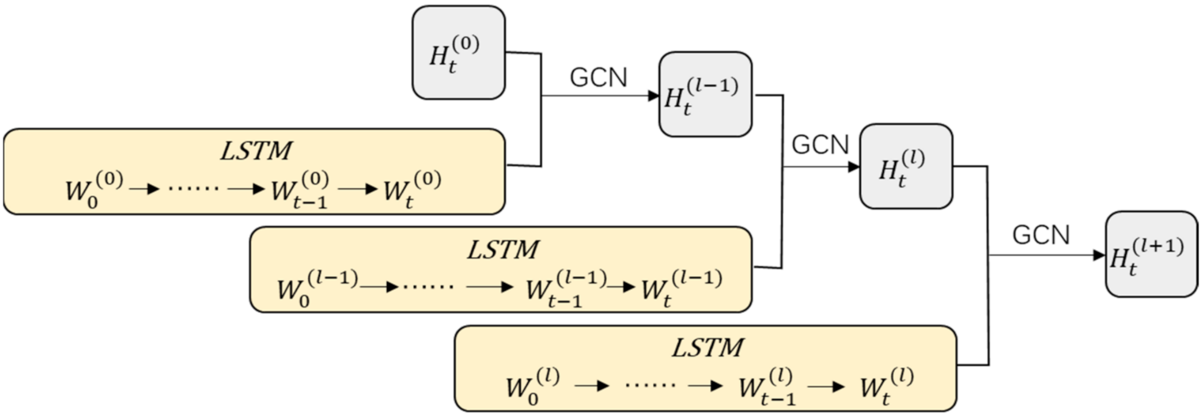

$H_t^{(0)}$
GCN
$H_t^{(l-1)}$
GCN
$H_t^{(l)}$
GCN
$H_t^{(l+1)}$
LSTM
$W_0^{(0)}$ …… $W_{t-1}^{(0)}$ $W_t^{(0)}$
LSTM
$W_0^{(l-1)}$ …… $W_{t-1}^{(l-1)}$ $W_t^{(l-1)}$
LSTM
$W_0^{(l)}$ …… $W_{t-1}^{(l)}$ $W_t^{(l)}$

Figure 5. Node update.

5. Experiment

5.1. Data Set and Processing

To verify the effectiveness of our model, we used the California highway dataset. PEMS uses sensors to acquire real-world traffic data from more than 8100 locations on California highways and highway systems, which are integrated into multiple time intervals. We selected the PEMS04 dataset and the PEMS08 dataset. The PEMS04 dataset contains the traffic data of San Francisco Bay from 1 January 2018 to 28 February 2018 collected by 3848 sensors, including three aspects of traffic, speed, and occupancy, where we selected data from 307 of these sensors for verification. The PEMS08 dataset contains the traffic data of San Bernardino from 1 July 2016 to 31 August 2016 collected by 1979 sensors, including three aspects of traffic, speed, and occupancy, where we selected data from 170 of these sensors for verification.

We first removed redundant sensors with distances of less than 3.5 miles; some data were missing from the original traffic speed dataset due to equipment failures, etc. Considering the spatiotemporal characteristics of traffic data, we used linear interpolation for missing values.

The traffic information in both datasets was updated every 5 min. In chronological order, we selected the first 60% of the data as the training set, the middle 20% of the data as the validation set, and the last 20% of the data as the test set.

Since the distance between each sensor was different, we chose the inverse of the distance as the element value of the adjacency matrix, thereby constructing the adjacency matrix. Because of the different dimensions, we normalized all the data, as shown in Equation (13).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{13}$$

5.2. Experimental Setup

Considering the influence of periodicity on the experimental results, we divided the experimental data into adjacent time series, daily period series, and weekly period series. They are represented by χ_h, χ_d, and χ_w, respectively. We fed χ_h, χ_d, and χ_w as inputs to the three LSTM-GCN subnetworks for training, respectively, and combined the outputs of the three subnetworks into the final output. We conducted experiments on a server configured with a Xeon Platinum 8163 processor clocked at 2.7 GHz and an NVIDIA Tesla P100 graphics card with 16 GB of VRAM. When training on the PMES04 dataset, the number of iterations was 100, the batch size was 16, and the Adam optimizer was used to update the parameters with a learning rate of 0.01. When training on the PMES08 dataset, the number of iterations was 200, the batch size was 32, the Adam optimizer was used for parameter update, and the learning rate was 0.01.

5.3. Evaluation Indicators

The experiment tests the model performance through *RMSE* (root-mean-square error), *MAE* ((mean absolute error) and *MAPE* (mean absolute percentage error); the formulas are defined as follows:

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n}|\hat{y}_i - y_i|, \tag{14}$$

$$MAPE = 100\% \times \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{15}$$

$$\text{and } RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{16}$$

where n is the number of predicted values, $\hat{y}_i$ is the predicted value, and y_i is the true value.

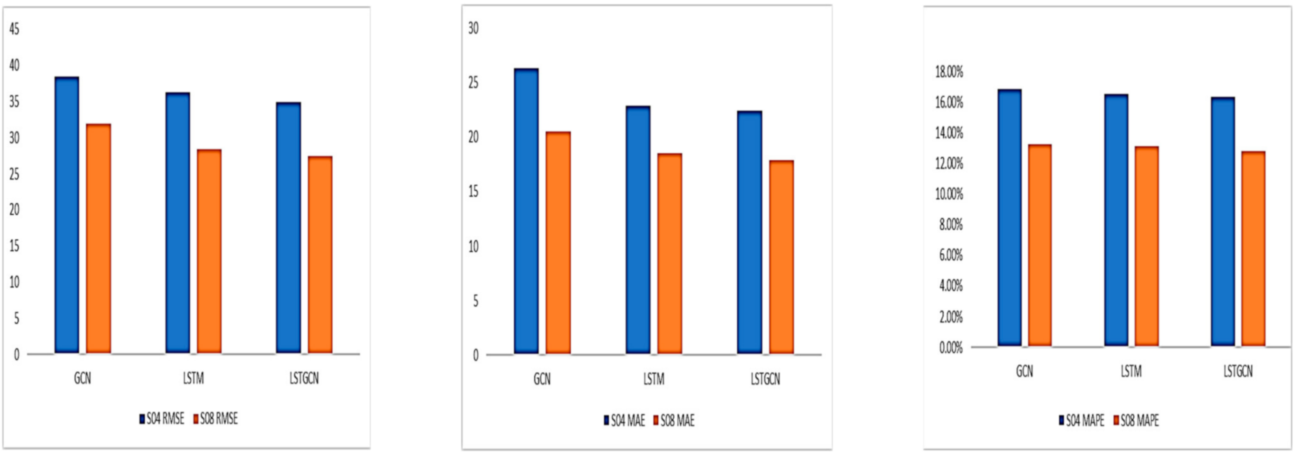
5.4. Results

As shown in Table 1, our model outperforms other models on both datasets. Since the HA model and the ARIMA model are linear models and only consider the information of the time dimension, the prediction effect of the models is relatively poor. The SVR model and the GRU model use machine-learning methods to analyze data, and have better nonlinear mapping capabilities than the HA model and the ARIMA model. However, the SVR model and the GRU model also only analyze the data from the time dimension, without considering the spatial dimension, so the prediction effect of the model is only better than the HA model and the ARIMA model. The ASTGCN model uses an attention mechanism from the temporal and spatial dimensions, respectively. Compared with the ARIMA model, the LSTM model, and the GRU model, the model considers the information of the spatial dimension, thereby significantly improving the prediction effect of the data. The LST-GCN model uses the LSTM model to update the parameters of the GCN model, which avoids the problem of too many parameters caused by separating the two models. It also considers the information of the time dimension and the space dimension. At the same time, the model also combines adjacent sequences and daily sequences. Three sequences of weekly sequence are used to predict the traffic flow. Considering the influence of periodicity on the prediction results, the data information is greatly utilized. Therefore, the model in this paper has achieved better prediction results than other models. For example, for the PEMS04 dataset, using RMSE, MAE, and MAPE as evaluation metrics, respectively, LST-GCN has an average improvement of 0.9%, 2.2%, and 1.3% compared with ASTGCN. For the PEMS08 dataset, using RMSE, MAE, and MAPE as evaluation metrics, respectively, LST-GCN achieves an average improvement of 2.5%, 3.7%, and 1.8% compared to ASTGCN.

Table 1. Average performance comparison of different approaches on PEMS04 and PEMS08.

Model	PMES04			PMES08		
	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)
HA	54.16	36.68	19.69	44.06	29.46	15.25
ARIMA	68.16	32.01	19.17	43.31	24.05	14.34
SVR	45.75	29.45	17.09	36.98	23.13	13.81
GRU	45.16	28.64	16.27	35.96	22.25	13.03
ASTGCN	35.23	22.93	16.58	28.16	18.61	13.05
LST-GCN	**34.93**	**22.43**	**16.37**	**27.47**	**17.93**	**12.81**

To confirm the spatiotemporal prediction ability of the LST-GCN model, we respectively compared the LST-GCN model with the LSTM model and the GCN model. As shown in Figure 6, our LST-GCN model has a strong spatiotemporal prediction ability. Since the LSTM model only considers the impact of time factors on traffic flow, while the GCN model only considers the impact of spatial factors on traffic flow, these two models cannot fully consider the information of the data. Therefore, the prediction accuracy of the LSTM model and GCN model is relatively poor. For example, using RMSE as the evaluation metric, on the PEMS04 dataset, LST-GCN has an average improvement of 9.2% compared with GCN, and an improvement of 3.4% compared to LSTM. On the PEMS08 dataset, LST-GCN has an average improvement of 13.9% compared to GCN and 3.5% compared to LSTM.

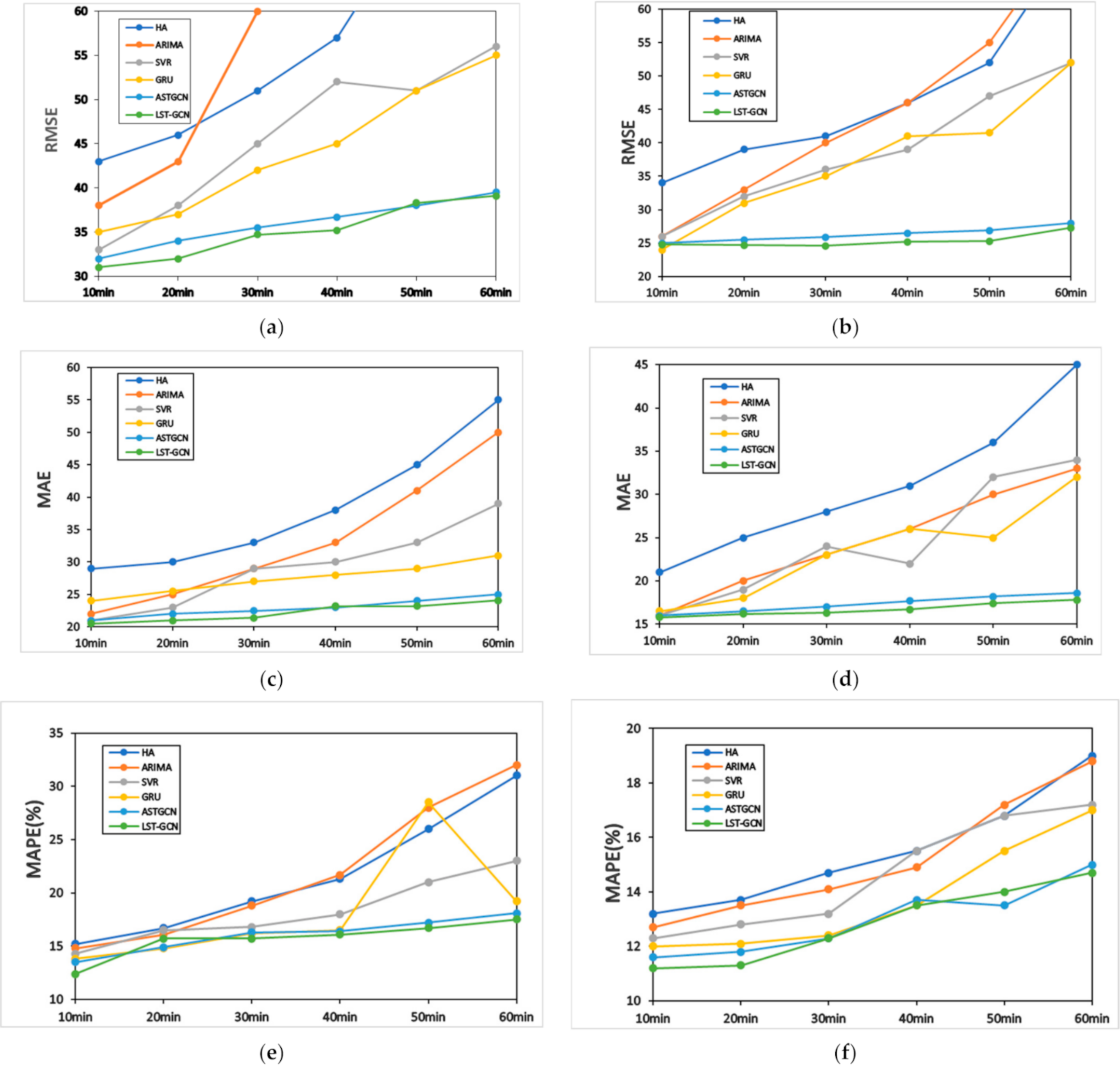


(**a**) (**b**) (**c**)

Figure 6. Average performance comparison of LST-GCN and GCN and LSTM on PEMS04 and PEMS08. (**a**) RMSE comparison of LST-GCN and GCN and LSTM on PEMS04 and PEMS08. (**b**) MAE comparison of LST-GCN and GCN and LSTM on PEMS04 and PEMS08. (**c**) MAPE comparison of LST-GCN and GCN and LSTM on PEMS04 and PEMS08.

Figure 7 shows how the prediction performance of the model varies with the range of prediction. With the increase in the prediction interval, the prediction error of the model will gradually increase, and the prediction effect will inevitably deteriorate. The RMSE, MAE, and MAPE values of the four models, HA, ARIMA, SVR, and GRU, increase continuously with the increase in prediction time, and the variation range is large. Compared with these four models, the ASTGCN model and the LST-GCN model continue to increase with the prediction time, but the variation range is relatively small. This is because the first four models only consider the impact of variation of time on the prediction results. With the increase in prediction interval, the time dimension information between roads on future traffic will have less and less impact, resulting in a lower and lower prediction accuracy of the model. In the long-term prediction, the spatiotemporal correlation is a more important predictor, so ASTGCN model and LST-GCN model are far superior to the other four models in the longer-term prediction. It can also be seen from the figure that the overall prediction effect of our LST-GCN model is better than that of ASTGCN model, which indicates that our LST-GCN model can better mine the spatiotemporal correlation of traffic data, to make more accurate predictions.

To better understand the LST-GCN model, we selected a road segment on the PEMS04 dataset and PEMS08 dataset, respectively, and visualized the prediction results on the test set. Figure 8a,b show the visualization results on two datasets, PEMS04 and PEMS08, respectively. It can be seen that the simulation effect of the model is better. It can be seen from the results that the prediction results of the LST-GCN model are relatively smooth. We speculate that it may be because the GCN model adds a smoothing filter to the Fourier domain and moves the filter to capture spatial features. This results in smoother experimental results.



(a)

(b)

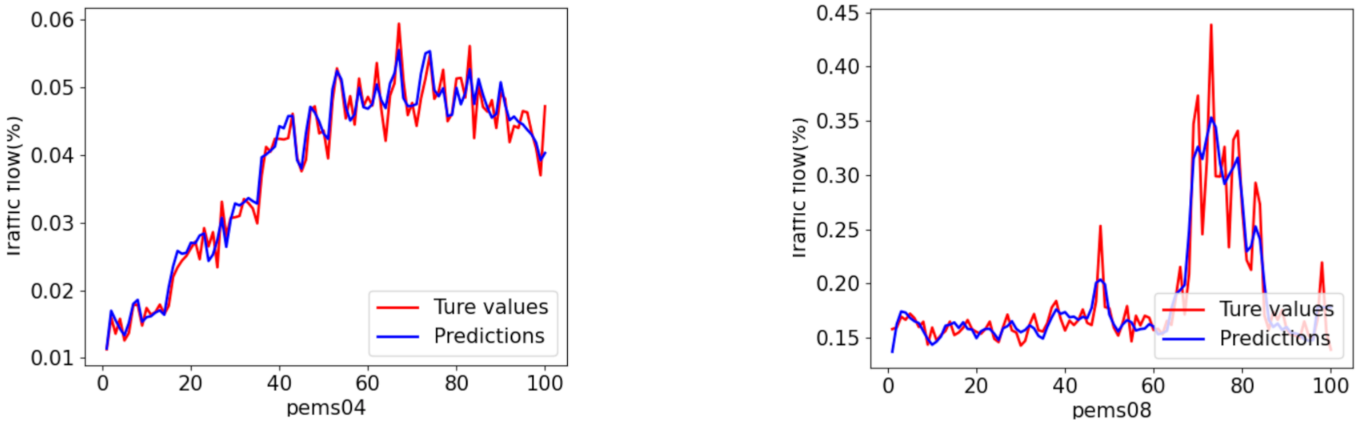
(c)

(d)

(e)

(f)

Figure 7. Performance changes of different methods as the forecasting interval increases. (**a**) Changes on PEMS04 dataset, based on RMSE. (**b**) Changes on PEMS08 dataset, based on RMSE. (**c**) Changes on PEMS04 dataset, based on MAE. (**d**) Changes on PEMS08 dataset, based on MAE. (**e**) Changes on PEMS04 dataset, based on MAPE. (**f**) Changes on PEMS08 dataset, based on MAPE.



(**a**) (**b**)

Figure 8. The visualization results for prediction. (**a**) Results on PEMS04 dataset. (**b**) Results on PEMS08 dataset.

6. Discussion

Accurate and rapid traffic flow prediction is an important issue affecting the development of intelligent transportation. The original traffic prediction model basically has the problem of large-parameter data or an inability to make full use of the data information. The reason why our model results are better than other models is mainly because of the following advantages: (1) We propose a new LST-GCN structure, which directly embeds the LSTM model into the updating process of GCN parameters, reducing the number of parameters; (2) compared with the model with a single model structure, our model considers both time and space factors, and makes full use of data information.

Our model improves the performance of short-term traffic flow, but there are still some issues to consider. Considering the "memory" capability introduced by the LSTM model may have a negative impact on the time complexity. [32–34] This effect exists in many cyclic structures. This needs further research in future work.

7. Conclusions

According to the traffic flow prediction problem, this paper proposes a method to update the model parameters of the graph convolutional network model using the long short-term memory neural-network model. By embedding the long short-term memory neural network into the graph convolutional network and modeling from the perspective of time and space at the same time, we further explore the internal connection of time and space. At the same time, three sequences of adjacent sequence, daily sequence, and weekly sequence are combined to predict traffic flow, and the influence of periodicity on the prediction result is considered. Finally, the method in this paper is compared with several common methods for predicting traffic flow through three evaluation indicators—RMSE, MAE, and MAPE—and it is concluded that the model proposed in this paper is better than other models on the PEMS dataset.

In the future, the main directions that need to be studied are: (1) applying the LST-GCN model to more road segments and increasing the prediction period of the model; (2) considering more complex road conditions, and improving our model by taking into account other factors such as weather and traffic accidents; (2) applying the LST-GCN model to other scenarios such as air quality prediction, energy prediction, etc.

Author Contributions: Conceptualization, X.H. and S.G.; Methodology, X.H.; Formal Analysis, S.G.; Writing—Original Draft Preparation, X.H.; Writing—Review & Editing, S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

GCN	Graph convolutional network
LSTM	Long short-term memory network
ARIMA	Autoregressive integrated moving average model
HA	History average model
ES	Exponential smoothing model
KF	Kalman filter model
SVR	Support vector regression model
KNN	K-nearest neighbor model
GRU	Gated recurrent unit model
CNN	Convolutional neural network
RNN	Recurrent neural network

References

1. Hani, S.M. Traveler behavior and intelligent transportation systems. *Transp. Res. Part C Emerg. Technol.* **1999**, *7*, 73–74.
2. Li, Y.; Lin, Y.; Zhang, F. Research on geographic information system intelligent transportation systems. *Chung-Kuo K. Lu Hsueh Pao China J. Highw. Transp.* **2000**, *13*, 97–100.
3. Liu, J.; Guan, W. A summary of traffic flow forecasting methods. *J. Highw. Transp. Res. Dev.* **2004**, *3*, 82–85.
4. Ma, X.; Tao, Z.; Wang, Y. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp. Res. C Emerg. Technol.* **2015**, *54*, 187–197. [CrossRef]
5. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2016**, arXiv:1609.02907.
6. Levin, M.; Tsao, Y.-D. On forecasting freeway occupancies and volumes. *Transp. Res. Rec.* **1980**, *773*, 47–49.
7. 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]
8. Shi, G.; Guo, J.; Huang, W.; Williams, B.M. Modeling seasonal heteroscedasticity in vehicular traffic condition series using a seasonal adjustment approach. *J. Transp. Eng.* **2014**, *140*, 5. [CrossRef]
9. Chan, K.Y.; Dillon, T.S.; Singh, J.; Chang, E. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 644–654. [CrossRef]
10. Kumar, S.V. Traffic flow prediction using Kalman filtering technique. *Procedia Eng.* **2017**, *187*, 582. [CrossRef]
11. Zhou, T.; Jiang, D.; Lin, Z.; Han, G.; Xu, X.; Qin, J. Hybrid dual Kalman filtering model for short-term traffic flow forecasting. *IET Intell. Transp. Syst.* **2019**, *13*, 1023–1032. [CrossRef]
12. Cai, L.; Zhang, Z.; Yang, J.; Yu, Y.; Zhou, T.; Qin, J. A noise-immune Kalman filter for short-term traffic flow forecasting. *Phys. A Stat. Mech. Appl.* **2019**, *536*, 122601. [CrossRef]
13. Zhang, S.; Song, Y.; Jiang, D.; Zhou, T.; Qin, J. Noise-identified Kalman filter for short-term traffic flow forecasting. In Proceedings of the IEEE 15th International Conference on Mobile Ad-Hoc Sensor Networks, Shenzhen, China, 11–13 December 2019; pp. 1–5.
14. Castro-Netoa, M.; Jeong, Y.S.; Jeong, M.K.; Hana, L. Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Syst. Appl.* **2009**, *36*, 6164–6173. [CrossRef]
15. Chang, H.; Lee, Y.; Yoon, B.; Baek, S. Dynamic near-term traffic flow prediction: System-oriented approach based on past experiences. *IET Intell. Transp. Syst.* **2012**, *6*, 292–305. [CrossRef]
16. Su, Z.; Liu, Q.; Lu, J.; Cai, Y.; Jiang, H.; Wahab, L. Short-time traffic state forecasting using adaptive neighborhood selection based on expansion strategy. *IEEE Access* **2018**, *6*, 48210–48223. [CrossRef]
17. Yin, H.; Wong, S.C.; Xu, J.; Wong, C.K. Urban traffic flow prediction using a fuzzy-neural approach. *Transp. Res. Part C* **2002**, *10*, 85–98. [CrossRef]
18. Çetiner, B.G.; Sari, M.; Borat, O. A Neural Network Based Traffic-Flow Prediction Model. *Math. Comput. Appl.* **2010**, *15*, 269–278. [CrossRef]
19. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, Wuhan, China, 11–13 November 2016.
20. Wu, Y.; Tan, H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *arXiv* **2016**, arXiv:1612.01022.
21. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 Apr–3 May 3 2018.
22. Sun, S.; Wu, H.; Xiang, L. City-Wide Traffic Flow Forecasting Using a Deep Convolutional Neural Network. *Sensors* **2020**, *20*, 421. [CrossRef]
23. Zhao, L.; Song, Y.; Zhang, C. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp.* **2020**, *21*, 3848–3858. [CrossRef]

24. Geng, X.; Li, Y.; Wang, L. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. *AAAI Conf. Artif. Intell.* **2019**, *33*, 3656–3663. [CrossRef]
25. Diao, Z.; Wang, X.; Zhang, D. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. *AAAI Conf. Artif. Intell.* **2019**, *33*, 890–897. [CrossRef]
26. Huang, R.; Huang, C.; Liu, Y. Lsgcn: Long short-term traffic prediction with graph convolutional networks. *Int. Joint Conf. Artif. Intell.* **2020**, 2355–2361.
27. Lv, M.; Hong, Z.; Chen, L. Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3337–3348. [CrossRef]
28. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. *AAAI Conf. Artif. Intell.* **2019**, *33*, 922–929. [CrossRef]
29. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral networks and locally connected networks on graphs. *arXiv* **2014**, arXiv:1312.6203.
30. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2016.
31. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
32. Mauro, M.D.; Galatro, G.; Liotta, A. Experimental Review of Neural-Based Approaches for Network Intrusion Management. *IEEE Trans. Netw. Service Manag.* **2020**, *17*, 2480–2495. [CrossRef]
33. Dong, S.; Xia, Y.; Peng, T. Network Abnormal Traffic Detection Model Based on Semi-Supervised Deep Reinforcement Learning. *IEEE Trans. Netw. Service Manag.* **2021**, *18*, 4197–4212. [CrossRef]
34. Pelletier, C.; Webb, G.I.; Petitjean, F. Deep Learning for the Classification of Sentinel-2 Image Time Series. In Proceedings of the IGARSS 2019, Yokohama, Japan, 31 July 2019.