


## Article

# Traffic Forecasting Based on Integration of Adaptive Subgraph Reformulation and Spatio-Temporal Deep Learning Model

Shi-Yuan Han <sup>\*</sup>, Qi-Wei Sun, Qiang Zhao, Rui-Zhi Han and Yue-Hui Chen

Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, Jinan 250022, China; sunqwjimson@gmail.com (Q.-W.S.); zq9701@gmail.com (Q.Z.); ise\_hanrz@ujn.edu.cn (R.-Z.H.); yhchen@ujn.edu.cn (Y.-H.C.)

<sup>\*</sup> Correspondence: ise\_hansy@ujn.edu.cn; Tel.: +86-531-8276-6503

**Abstract:** Traffic forecasting provides the foundational guidance for many typical applications in the smart city management, such as urban traffic control, congestion avoidance, and navigation guidance. Many researchers have focused on the spatio-temporal correlations under fixed topology structure in traffic network to improve the traffic forecasting accuracy. Despite their advantages, the existing approaches are not completely discussed that the association relationship among traffic network nodes are not invariable under different traffic conditions. In this paper, a novel traffic forecasting framework is proposed by integrating the dynamic association of traffic nodes with the spatio-temporal deep learning model. To be specific, an adaptive subgraph reformulation algorithm is designed first based on the specific forecasting interval to reduce the interference of irrelevant spatio-temporal information. After that, by enhancing the attention mechanism with the generative decoder, a spatio-temporal deep learning model with only one forward operation is proposed to avoid the degradation of accuracy in the long-term prediction, in which the spatio-temporal information and the external factors (such as weather and holiday) are fused together to be as an input vector. Based on the reformulated subgraph constructed of traffic nodes with closer spatio-temporal correlation, experiments show that the proposed framework consistently outperforms other GNN (Graph Neural Network)-based state-of-the-art baselines for various forecasting intervals on a real-world dataset.

**Keywords:** traffic forecasting; spatial-temporal data; adaptive subgraph reformulation; encoder-decoder



**Citation:** Han, S.-Y.; Sun, Q.-W.; Zhao, Q.; Han, R.-Z.; Chen, Y.-H. Traffic Forecasting Based on Integration of Adaptive Subgraph Reformulation and Spatio-Temporal Deep Learning Model. *Electronics* **2022**, *11*, 861. <https://doi.org/10.3390/electronics11060861>

Academic Editor: Dionysios Kehagias

Received: 1 February 2022

Accepted: 3 March 2022

Published: 9 March 2022

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



**Copyright:** © 2022 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

As the increasing complexity of urban transportation, many social problems are becoming more and more serious such as air pollution and traffic congestion. As one of the most important infrastructure facilities in modern cities, ITSs (Intelligent Transportation Systems) provide the great conveniences in daily travel [1]. Intervention in advance based on traffic forecasting is regarded as a key for improving the efficiency of transportation, where the traffic condition is mainly detected by sensors installed on roads, subways and buses. However, there are complex spatial dependencies among sensors distributed on a road network, even the nearest two sensors in Euclidean space. Meanwhile, mass traffic data are often with high dimension and affected by emergencies, such as traffic accident and holidays. Therefore, the major challenge for traffic forecasting is that the spatial-temporal correlation must be captured simultaneously to improve the traffic forecasting accuracy.

The early methods of traffic forecasting are usually based on historical traffic data. Traditional statistical analysis methods with regression functions are adopted to analyze the historical traffic data, such as linear regression prediction method and ARIMA (Auto-Regressive Integrated Moving Average) method [2]. The most statistical analysis methods are simple and easy to be implemented under the assumption that the traffic time-series data is linearly modeled. However, due to the high nonlinearity of traffic data, statistical

analysis models are not completely capture the changing trends of traffic flow in the next period. Meanwhile, the traditional machine learning methods rely on manually extracted traffic features under the simple traffic conditions or small volume of traffic data. Recently, deep learning methods have been widely used in various complex prediction tasks to improve the forecasting accuracy, such as feedforward neural networks [3], deep belief networks [4] and RNN (Recurrent Neural Networks) [5]. Specially, ref. [6] uses LSTM (Long Short-Term Memory) network for traffic speed forecasting; [7] combines LSTM and Gaussian process regression to forecast the traffic load of network. However, the above mentioned methods are only adopted to study the temporal characteristics of mass traffic data.

Inspired by image feature extraction, CNNs (Convolutional Neural Networks) are used to extract spatial features of road network [8,9], in which the traffic network and the irregular road structure are converted into the image grid data with fixed adjacency relationship. However, the spatial correlation among traffic nodes will be destroyed. To overcome the shortcomings of CNNs, GNNs (Graph Neural Networks) are employed to predict the traffic flow on non-grid road networks [10–13], in which a non-Euclidean structure is used to capture the spatial correlation based on the adjacency matrix. Recently, various enhanced GNNs have been designed for traffic forecasting under the complex, non-linear and spatio-temporal correlated traffic networks [14–18]. Specially, ref. [14] proposes an optimized graph convolutional recurrent neural network to learn an optimized graph in a data-driven manner and to reveal the potential relationships between the road segments with the traffic data. Ref. [15] introduces an attention mechanism based on DCRNN (Diffusion Convolutional Recurrent Neural Network) to obtain the adaptive weight matrix of nodes, in which the weight matrix assigned to the nodes by graph convolution is fixed. Those studies have fully demonstrated the effectiveness of GNN-based models to reveal the spatio-temporal correlation on complex traffic network. However, their common limitations have been recognized as follows:

- (1) No matter what the forecasting intervals and the traffic condition, most existing models often require the extraction of spatial features on the entire traffic network with fixed nodes and traffic network topology. However, the association relationships among traffic nodes are not invariable.
- (2) The traffic forecasting methods based on deep learning models are with the forms of step by step, such as LSTM, Seq2Seq (Sequence to Sequence) model, and Transformer. While dealing with long sequence of traffic data, the prediction accuracy will deteriorate caused by the aggravated propagation of accumulated errors.

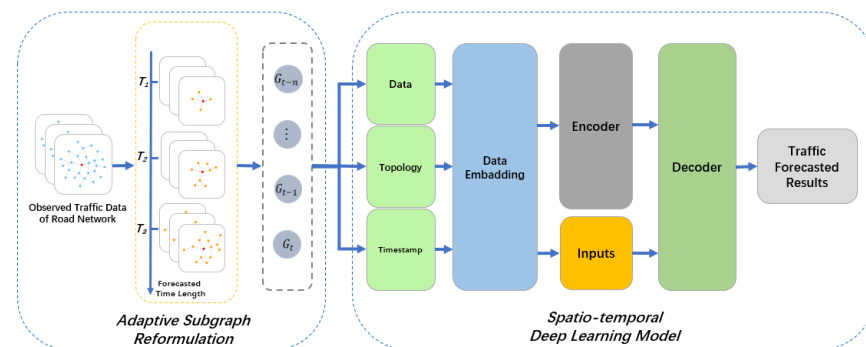
Motivated by the above discussion, by integrating the dynamic association of traffic nodes with the enhanced attention mechanism, a traffic forecasting framework with adaptive subgraph reformulation is proposed to improve forecasting accuracy involving with different forecasting intervals. The main contributions of this research are as follows:

- (1) An algorithm of adaptive subgraph reformulation is proposed to reduce the impact of irrelevant spatiotemporal information on the forecast accuracy, in which the traffic nodes and network topology in traffic network are selected and reconstructed adaptively based on the reachability analysis and similarity quantization among traffic nodes within a specific forecasting interval.
- (2) A spatio-temporal deep learning model with self-attention mechanism is designed to avoid the cumulative error propagation caused by the long sequence of traffic data. The fusion of the time-series information, the spatial information and the time-stamp information sets as the input, and all forecasting results are output by performing one forward operation based on the generative decoder.

The framework of this paper is as follows: In Section 2, the proposed traffic forecasting framework is presented. The corresponding experimental results are given in Section 3. This paper will end with the conclusion and future work in Section 4.

## 2. Methodology

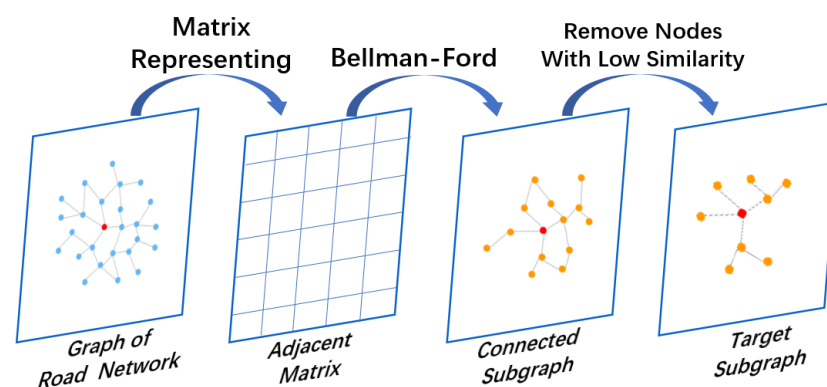
The traffic forecasting framework with adaptive subgraph reformulation is shown in Figure 1, which contains two main components: the adaptive subgraph reformulation algorithm and the spatio-temporal deep learning model. In adaptive subgraph reformulation module, the number of nodes and network topology in traffic network are constructed based on the forecasting interval. Based on the reformulated network topology, the spatio-temporal deep learning model is intended to fuse the spatio-temporal data with the external factors and then use the encoder–decoder to predict the traffic flow.



**Figure 1.** The structure of traffic forecasting framework with adaptive subgraph reformulation.

### 2.1. Adaptive Subgraph Reformulation Algorithm

Different from the pre-defined static graph adapted in GCN model, an effective subgraph is reformulated to reduce the interference of irrelevant spatio-temporal information by using the designed adaptive subgraph reformulation algorithm. First, the adjacent matrix of the maximum traveling time from the original graph of traffic network is constructed. Then, a temporary connected subgraph is obtained by using the shortest-path algorithm, in which the nodes are selected based on the reachability analysis and similarity quantization in the specific forecasting interval. Finally, by employing the spectral clustering on the Laplacian matrix of the temporary connected subgraph, the target subgraph is obtained by remove nodes with low spatio-temporal similarity. The specific process of adaptive subgraph reformulation is shown in Figure 2.



**Figure 2.** The process of the adaptive subgraph reformulation.

Leong [19] observes that the speed distribution of vehicles on the road obeys a normal distribution. He has obtained the results after using radar speedometers to measure the traffic speed of smooth flow at 31 observation points in Australia during a three-year period. Katti and Raghavachari [20] have found that the speeds of cars, heavy commercial vehicles, light commercial vehicles and motorcycles, all follow the normal distribution. Kumar and Rao [21] have also demonstrated that the vehicle speed obeys a normal distribution.

First, under the assumption that the vehicle speed satisfies the normal distribution  $N(\mu, \sigma^2)$ , the minimum speeds among different traffic nodes could be computed. In details,

the average vehicle speed during the latest  $t$  hours is brought into the normal distribution as  $\mu$ . 120% of the maximum speed limit is used as the upper bound of the 95% of the confidence interval  $(\mu + 2\sigma)$ . The lower confidence bound is obtained as the minimum speed on the road by  $(\mu - 2\sigma)$ . Thus, the distance matrix  $D$  and minimum speed matrix  $S$  are constructed from the traffic dataset:

$$D = \begin{bmatrix} 0 & d_{12} & \cdots & \cdots & d_{1n} \\ d_{21} & 0 & & & d_{2n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & \cdots & 0 \end{bmatrix}, \quad S = \begin{bmatrix} 0 & s_{12} & \cdots & \cdots & s_{1n} \\ s_{21} & 0 & & & s_{2n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & \cdots & 0 \end{bmatrix}, \quad (1)$$

where  $n$  is the number of nodes in the overall traffic network;  $d_{ij}$  in  $D$  represents the actual distance from node  $i$  to node  $j$ ;  $s_{ij}$  in  $S$  represents the minimum speed from node  $i$  to node  $j$ . The maximum traveling time matrix  $T$  is obtained by dividing the corresponding elements of the  $D$  and  $S$  matrices, which is described as:

$$T = \begin{bmatrix} 0 & t_{12} & \cdots & \cdots & t_{1n} \\ t_{21} & 0 & & & t_{2n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & \cdots & 0 \end{bmatrix} = \begin{bmatrix} 0 & d_{12}/s_{12} & \cdots & \cdots & d_{1n}/s_{1n} \\ d_{21}/s_{21} & 0 & & & d_{2n}/s_{2n} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ d_{n1}/s_{n1} & d_{n2}/s_{n2} & \cdots & \cdots & 0 \end{bmatrix}, \quad (2)$$

where  $t_{ij}$  represents the maximum traveling time from node  $i$  to node  $j$ .

After that, by employing the shortest-path algorithm of Bellman–Ford method in [22], the traveling time  $T_{pass}^i$  from the  $i$ th node to the target forecasting node could be computed. The nodes with the traveling time shorter than the forecasting interval  $T_f$  will be selected to construct a temporary connected subgraph. Obviously, those dropped nodes will not affect the forecasting target node within the specific forecasting interval. The adjacent matrix  $G_T$  of the designed temporary connected subgraph will be constructed based on the selected nodes and the corresponding distances.

By applying the spectral clustering on the Laplacian matrix of the adjacent matrix  $G_T$  of the reformulated temporary connected subgraph, the similarity between the  $i$ th node to the target forecasting node could be quantized as  $H_s^i$ . Filtrating the nodes with less similarity quantitative value  $H_s^i$  than a reasonable threshold  $H$ , the reformulated subgraph could be obtained with the adjacent matrix  $G$ , which is described as

$$G = \begin{bmatrix} 0 & g_{12} & \cdots & \cdots & g_{1m} \\ g_{21} & 0 & & & g_{2m} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ g_{m1} & g_{m2} & \cdots & \cdots & 0 \end{bmatrix}, \quad (3)$$

where  $m$  is the number of nodes in the reformulated subgraph;  $g_{ij}$  represents the actual distance from node  $i$  to node  $j$ . Due to the difference in road connectivity,  $D$  is usually an asymmetric matrix.

Based on the above description, the designed adaptive subgraph reformulation Algorithm 1 is given as follows.

**Algorithm 1** Adaptive subgraph reformulation algorithm

**Require:** The actual distances between each traffic nodes; the maximum speed limits in each road segment; the forecasting interval  $T_f$ ; the threshold of similarity  $H$ .

- Obtain the initial distance matrix  $D$  and minimum speed matrix  $S$  from the dataset;
- Calculate the maximum travelling time matrix  $T$ ;
- Get the vector  $T_{pass} = [T_{pass}^1, T_{pass}^2, \dots, T_{pass}^i, \dots, T_{pass}^n]$  based on Bellman–Ford method, where  $T_{pass}^i$  denotes the shortest traveling time from the  $i$ th node to the forecasting target node;
- Construct the temporary connected subgraph with the adjacent matrix  $G_T$  by dropping those nodes with  $T_{pass}^i > T_f$  from original traffic network;
- Calculate the similarity vector  $H_s = [H_s^1, H_s^2, \dots, H_s^i, \dots, H_s^k]$  among the nodes in the temporary connected subgraph based on the spectral clustering;
- Construct the reformulated subgraph with adjacent matrix  $G$ , where the nodes with  $H_s^i < H$  are dropped in the temporary connected subgraph;
- Output: The adjacent matrix  $G$  of the reformulated subgraph.

## 2.2. Spatio-Temporal Deep Learning Model

There is an inherent limitation of the encoder–decoder architecture with Transformer in [23] that prevents it from being directly applicable to the time-series forecasting. To address this issue, we have changed the decoder’s structure. The generative style decoder, though conceptually simple, predicts the long time-series sequences in one forward operation rather than in a step-by-step way. The one forward operation will drastically improve the inference speed of the long-sequence predictions.

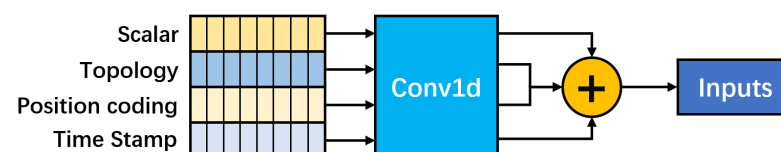
Referring to the structure of Transformer model, the proposed spatio-temporal deep learning model consists of three components, including the embedding module used to preprocess the traffic data, the encoder module used to capture the dependencies between input sequences, and the decoder module used to forecast the traffic flow through one forward calculation. What’s following, the designed three components will be described in detailed.

### 2.2.1. Embedding Module of Traffic Data

The embedding module of traffic data consists of four parts:

- **Scalar:** contains the traffic data of each nodes in the reformulated subgraph;
- **Network Topology:** contains the spatial characteristics derived from the obtained reformulated subgraph;
- **Position coding:** contains the traffic flow data with the position coding of each nodes in the reformulated subgraph;
- **Time Stamp:** contains the external information expressed as  $n$ -dimensional variable, including the information about year, month, day, hour, minute, week, weather, and holiday.

As shown in Figure 3, the embedding of traffic data expresses as a 512-dimensional vector converted from the above four parts, which is defined as the input of the proposed spatio-temporal deep learning model.



**Figure 3.** The process of embedding of traffic data.

### 2.2.2. Self-Attention Mechanism

The attention mechanism is first proposed in the field of natural language processing (NLP) [24,25], and gradually applied to the computer vision [26,27] and the time-series prediction [28,29]. As is well known, Recurrent Neural Networks (RNNs) for prediction with short sequences show good performance, in which all steps are encoded in the sequence and a final output of a sequence embedding is given out. In order to make up the long sequences, the proposed model in [25] uses all the hidden states of the encoder at the decoder end, and gives attention to certain hidden states when decoding each element. Specially, the self-attention mechanism is proposed and supposed to satisfy all requirements for the encode sequences, where each elements attend to every other element in the same sequence or other sequences.

In order to capture the spatio-temporal correlation among different traffic sequences in the reformulated subgraph, the self-attention mechanism and the multi-headed self-attention mechanism are employed and described as follows.

**Self-attention mechanism:** Based on the same input  $X$ , three matrices are designed to realize the self-attention mechanism, including the query of  $Q = XW^Q$ , the key  $K = XW^K$  and the value of  $V = XW^V$ , where  $W^Q$ ,  $W^K$  and  $W^V$  are the trained weight matrices, and the superscripts of  $W^Q$ ,  $W^K$  and  $W^V$  are the outputs obtained after the operation. In order to stabilize the gradient, attention function is designed by using *Softmax* function, which is described as

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad (4)$$

where  $d$  denotes the dimension of  $QK^T$ .

**Multi-headed self-attention mechanism:** In order to focus on different features of the traffic information, the deep learning model is divided into multiple heads to form the multiple subspaces. It means that the self-attention operation will be run  $N$  times simultaneously, where  $N$  is the number of heads. Meanwhile, the outputs of  $\{head_1, head_2, \dots, head_n\}$  obtained from the multi-head are stacked together. Then, by using the splicing function  $Concat(\bullet)$ , a linear transformation is designed to realize the multi-head attention, which is described as:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_n)W^O, \quad (5)$$

where  $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$  with  $i = 1, \dots, N$  denotes the attention of the  $i$ th head, in which  $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$ ,  $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ ,  $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ , and  $W^O \in \mathbb{R}^{dN_{model} \times d_v}$  are the trained weight matrices;  $d_{model}$  is the dimension of input vector;  $d_k = d_v = d_{model}/N$ . The structure of multi-headed self-attention mechanism is displayed in Figure 4.

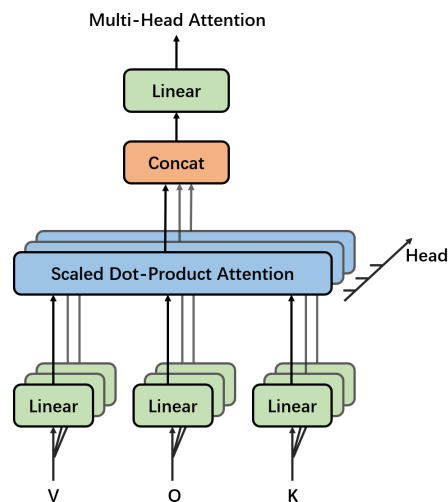


Figure 4. The structure of multi-head attention mechanism.



### 2.2.3. Encoder–Decoder Component

In the proposed spatio-temporal deep learning model, a self-attention mechanism is adopted in the encoder module. A multi-head self-attention is employed in the decoder module to obtain all prediction results by one forward operation. The structure of the encoder–decoder component is shown in Figure 5.

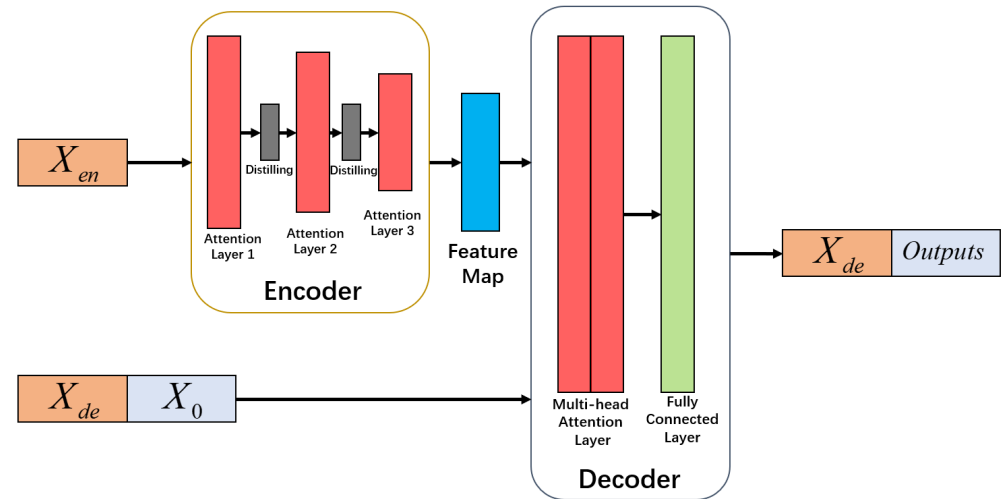


Figure 5. The structure of encoder–decoder component.

The inputs of the encoder–decoder component is shown in Figure 6.

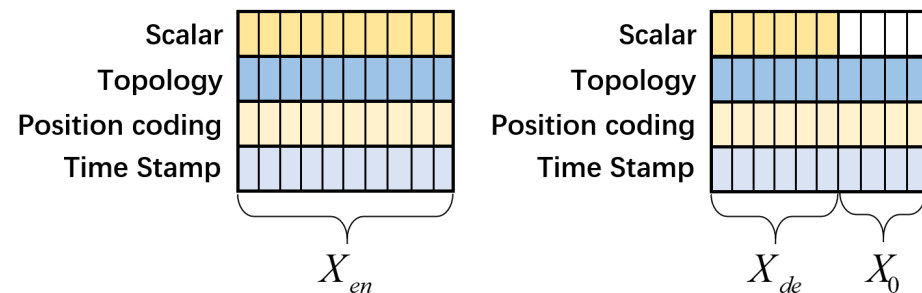


Figure 6. The inputs of the encoder–decoder component.

In order to extract the robust long-term dependency of the long sequential inputs, the input  $X_{en}$  of encoder module is designed as:

$$X_{en} = \{X_{en}^{t-n}, X_{en}^{t-n+1}, \dots, X_{en}^t\}, \quad (6)$$

where  $X_{en}^{t-i} \in \mathbb{R}^{d_{model} \times L_x}$  with  $i = 0, 1, \dots, n$  denotes the  $(t-i)$ th sequence input, in which  $L_x$  is the length of input. Meanwhile, the input of the decoder module is formulated as  $X_{input} = \{X_{input}^{t-n}, X_{input}^{t-n+1}, \dots, X_{input}^t\}$ , where the element  $X_{input}^{t-i}$  with  $i = 0, 1, 2, \dots, n$  is described as:

$$X_{input}^{t-i} = \text{Concat}(X_{de}^{t-i}, X_0^{t-i}) \in \mathbb{R}^{(L_{de}+L_0) \times d_{model}}, \quad (7)$$

in which  $X_{de}^{t-i} \in \mathbb{R}^{L_{de} \times d_{model}}$  is the start token;  $X_0^{t-i} \in \mathbb{R}^{L_0 \times d_{model}}$  is a placeholder for the target sequence;  $L_{de}$  and  $L_0$  represent the lengths of the start token and the placeholder, respectively.

Encoder module receives the embedding values of traffic data and outputs the Feature Map to the decoder module, which consists of a 3-layer encoder layer and a fully connected layer. More specially, the encoder layer mainly contains two sub-layers: a self-attention layer and a feedforward layer composed of two linear mappings. Both sub-layers are followed by a batch normalization layer. Between the sub-layers, the jump connections are

employed to avoid the excessive fitting. Between every two attention layers, the distilling operation is performed to halve the length dimension, which involves with 1-dimensional convolution, *ELU* activation function, and the maximum pooling. The distilling procedure forwarded from the  $j$ th layer into the  $(j + 1)$ th layer is expressed as:

$$X_{j+1}^t = \text{MaxPool}(\text{ELU}(\text{Conv1d}[X_j^t])), \quad (8)$$

where  $X_j^t \in \mathbb{R}^{d_{\text{model}} \times L_x}$  represents the  $t$ th sequence input in the  $j$ th layer;  $[\cdot]_{AB}$  represents the attention layer.

Decoder module employs a structure of generative style decoder [30] to perform one forward step on the time-series and output all the forecasting results, which includes two attention layers and a two-layer feedforward part of linear mapping. In the first attention layer, by multiplying the embedding input with the trained weight matrix of decoder module, the values of  $Q$ ,  $K$ , and  $V$  are obtained. Meanwhile, the triangular causal mask mechanism is employed in this layer to ensure that the output decoded at time  $t$  depends only on the output before time  $t$ . In the second attention layer, the value of  $Q$  is obtained by multiplying the output of the first attention layer; the values of  $K$  and  $V$  are obtained by multiplying the output of encoder.

Based on the above description, the proposed spatio-temporal deep learning model is described in Algorithm 2 in details.

---

**Algorithm 2** Spatio-temporal deep learning model

---

**Require:** The dataset of traffic flow data; the adjacent matrix  $G$  of the reformulated subgraph.

- Read the traffic flow data  $TF$ , the position information  $PE$ , and the time stamps  $TS$  from the traffic dataset;
  - Construct the network topology  $NT$  from  $G$  based on the reformulated subgraph;
  - Obtain the input  $X_{en}$  of the encoder module and  $X_{input}$  of the decoder module by embedding  $TF$ ,  $PE$ ,  $NT$ , and  $TS$ ;
  - Get the feature map  $FM$  based on the output of the encoder module;
  - Calculate the output of the decoder module based on  $X_{input}$  and  $FM$ ;
  - Return the Output of the decoder module.
- 

### 3. Experiments

#### 3.1. Dataset

The effectiveness of the proposed traffic forecasting framework is validated on a dataset collected by the Didi Chuxing GAIA Initiative (<https://GAIA.didichuxing.com> (accessed on 13 March 2021)) in Jinan, China. The employed dataset contains the average speed of vehicles collected from 61 road monitors in Jinan ranging from 1 January 2018 to 31 December 2018 for the experiment. The sample period sets as 10 min. After that, the length and the connection relationship of each traffic node are calculated based on the longitude and latitude coordinates of each traffic node in the traffic dataset.

For data preprocessing of the traffic dataset, the normalization with Z scores is employed first to standardize the traffic data input. The time stamp information of the traffic data set is with the 4-dimensional information, including [month, date, hour, minute]. Meanwhile, the time-stamp information distribution is in the range of  $[-0.5, 0.5]$ . Meanwhile, the dataset is divided into 60% for training, 20% for validation, and 20% for testing. The proposed traffic forecasting framework is trained on the training set and tested on the test set. It is worth noting that we filled in the missing values in the data set with the average speed of one hour before.



### 3.2. Experimental Settings and Results

Experiments are conducted under the environment with one Intel(R) Core(R) i9-9900K CPU @ 3.60 GHz (Intel, Santa Clara, CA, USA) and a NVIDIA GeForce 2080Ti GPU 11 GB card (NVIDIA, Santa Clara, CA, USA). The batch size is 32 and the training epoch is 100. The head size of the multi-head attention is 8. The Gaussian error linear units is defined as activation function. The learning rate sets as 0.001. *MSE* (Mean Squared Error) is defined as the loss function.

Comparison results with seven deep-learning traffic forecasting methods are employed to verify the advancements and effectiveness of the proposed framework under the same traffic dataset, including the HA (Historical Average), ARIMA, LSTM in [31], GRU in [32], GCRN in [33], Gated-STGCN in [34], and GWNET in [35]. To estimate the impact of adaptive subgraph reformulation on the graph, we make a version of our model without structure, i.e., only using spatio-temporal deep learning model, denoted by Ours\_WA. To further evaluate the efficiency of the proposed Spatio-Temporal Deep Learning Model, we replace the proposed deep learning model with Transformer, denoted by Ours\_WS. Considering the different forecasting intervals with 30 min, 60 min and 120 min, three performance indicators are measured under the above methods and the proposed framework, including *MAE* (Mean Absolute Error), *MAPE* (Mean Absolute Percent Error), and *RMSE* (Root Mean Square Error). More specially, the comparison results of the above three performance indicators are displayed in Tables 1–3. Meanwhile, in order to show the effectiveness of the proposed traffic forecasting framework more intuitively, the comparison curves between the forecasting results and the actual values are shown in Figures 7–9 under different forecasting intervals with 30 min, 60 min and 120 min, where the *x*-axis and *y*-axis represent the time step and the speed value, respectively.

From Tables 1–3, it could be noticed that the proposed traffic forecasting framework performs better than other prediction models in both the short-term and the long-term forecasting interval. In particular, as to the long-term traffic forecasting, the values of *MAE*, *RMSE* and *MAPE* are improved more markedly than other GNN-based state-of-the-art baselines in the short-term traffic forecasting. This demonstrates that the proposed traffic forecasting framework can capture the spatio-temporal correlations of urban road networks under different forecasting intervals effectively.

**Table 1.** The traffic forecasting results under different models with 30 min forecasting interval.

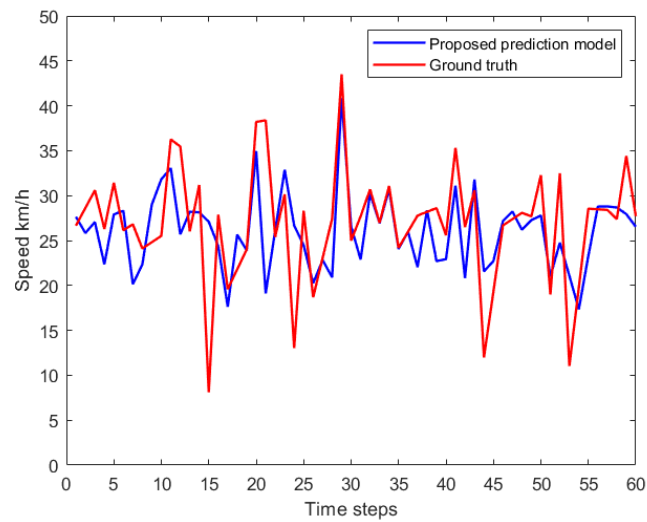
Method	HA	ARIMA	LSTM	GRU	GCRN	Gated-STGCN	GWNET	Ours_WA	Ours_WS	Ours
<i>MAE</i>	5.69	3.96	3.21	3.20	2.99	2.96	2.89	2.93	2.89	2.93
<i>RMSE</i>	7.60	6.14	4.85	4.85	4.54	4.48	4.37	4.48	4.37	4.13
<i>MAPE</i>	20.02%	14.12%	12.85%	12.82%	12.11%	11.85%	11.49%	11.88%	11.89%	11.84%

**Table 2.** The traffic forecasting results under different models with 60 min forecasting interval.

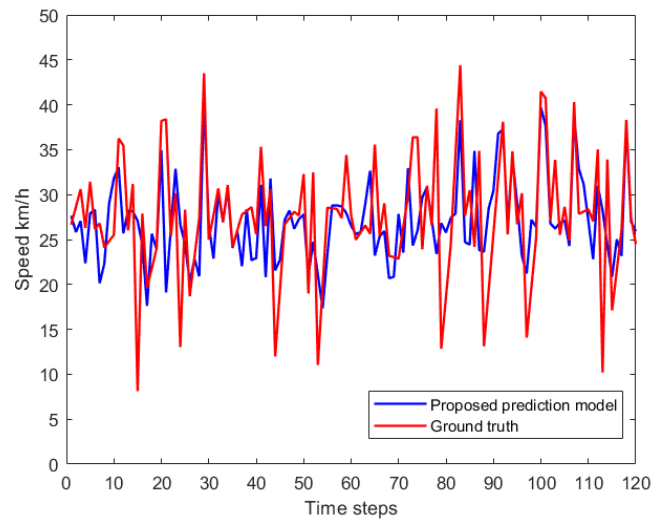
Method	HA	ARIMA	LSTM	GRU	GCRN	Gated-STGCN	GWNET	Ours_WA	Ours_WS	Ours
<i>MAE</i>	5.69	4.48	3.67	3.67	3.30	3.29	3.16	3.15	3.03	2.98
<i>RMSE</i>	7.60	6.56	5.48	5.47	4.96	4.92	4.71	3.29	3.16	4.09
<i>MAPE</i>	20.02%	16.10%	14.92%	14.86%	13.53%	13.33%	12.52%	11.58%	11.47%	11.38%

**Table 3.** The traffic forecasting results under different models with 120 min forecasting interval.

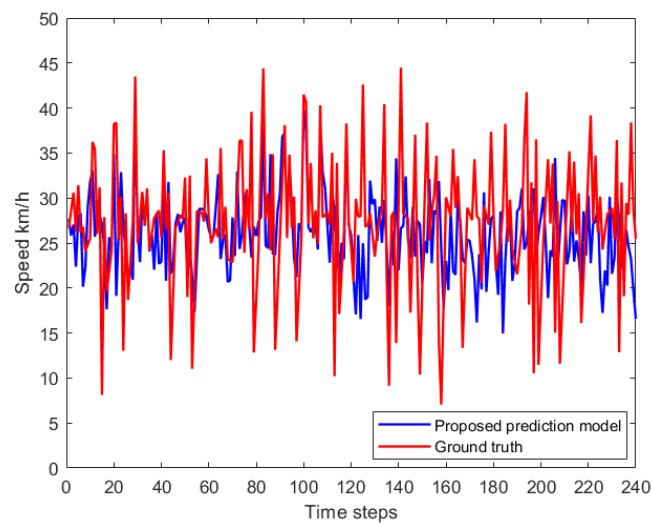
Method	HA	ARIMA	LSTM	GRU	GCRN	Gated-STGCN	GWNET	Ours_WA	Ours_WS	Ours
<i>MAE</i>	5.69	5.09	4.30	4.30	3.71	3.71	3.52	3.30	3.34	3.29
<i>RMSE</i>	7.60	7.01	6.26	6.26	5.45	5.44	5.16	5.05	4.99	4.85
<i>MAPE</i>	20.02%	18.24%	17.38%	17.41%	15.13%	15.03%	13.77%	13.25%	12.63%	12.36%



**Figure 7.** The comparison covers under 30 min forecasting interval.



**Figure 8.** The comparison covers under 60 min forecasting interval.



**Figure 9.** The comparison covers under 120 min forecasting interval.

To estimate the effect of adaptive subgraph reformulation and spatio-temporal deep learning model, we design the validation experiments and show the forecasting results of Ours\_WA and Ours\_WS in Tables 1–3. From the results, we can conclude that both of adaptive subgraph reformulation and spatio-temporal deep learning model of the proposed traffic forecasting framework are effective as the results outperform both of Ours\_WA and Ours\_WS.

Particularly, it is necessary to point out that the original dataset includes 561 traffic nodes. This means that a graph consisting of 561 nodes needs to be constructed for training the traditional GNN-based prediction models. However, while applying the designed adaptive subgraph reformulation algorithm, the nodes in reformulated subgraph are 82, 164, and 220 under different forecasting intervals with 30 min, 60 min, and 120 min, respectively. The number of nodes will be reduced by 85.38%, 70.77%, 60.78%. From Tables 1–3 and Figures 7–9, with relatively few traffic nodes, the forecasting accuracies still retain a satisfactory level. On the contrary, the proposed traffic forecasting framework with adaptive subgraph reformulation outperforms the employed state-of-the-art traffic forecasting models. This means that, on the premise of not reducing the forecasting accuracy, the adaptive subgraph reformulation algorithm can reduce the amount of input data under the dynamic urban traffic flow effectively.

From the above analysis, under different forecasting intervals, the proposed traffic forecasting framework performs better for improving the forecasting accuracy with less traffic information.

#### 4. Conclusions

In this paper, we proposed a new traffic forecasting framework to improve the forecasting performance on traffic road network under different forecasting intervals, which consisted of the adaptive subgraph reformulation algorithm and the spatio-temporal deep learning model. Based on the accessibility evaluation and the similarity quantification method among nodes in traffic network, the designed adaptive subgraph reformulation algorithm reduced the interference of irrelevant spatio-temporal information, in which the traffic nodes and their links with the weight were adaptively selected to reformulate the traffic network topology. Meanwhile, different from the step-by-step prediction in traditional deep-learning models, the designed spatio-temporal deep learning model with only one forward operation could obtain all the prediction results, where the attention mechanism was enhanced by applying the generative decoder. However, considering the dynamic complexity of the road network and the interference of weather or other factors, more data sources should be introduced in the traffic forecasting in the future.

For future work, we will integrate different GCN (Graph Convolutional Network)-based deep-learning models for fusing the captured features either the short-term traffic forecasting or the long-term traffic forecasting, such as GCN with LSTM for short-term traffic forecasting, GCN with Seq2Seq for middle-term traffic forecasting, and GCN with In-former for long-term traffic forecasting. In addition, the occasion of subgraph reformulation could be discussed with traffic factors, such as traffic congestion and accident.

**Author Contributions:** Methodology, S.-Y.H.; Supervision, S.-Y.H. and Y.-H.C.; Validation, Q.Z.; Writing—original draft, Q.-W.S.; Writing—review and editing, R.-Z.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grants 61903156 and 61873324, the Natural Science Foundation of Shandong Province for Key Project under Grant ZR2020KF006, the Natural Science Foundation of Shandong Province under Grant ZR2019MF040, the University Innovation Team Project of Jinan under Grant 2019GXRC015, the Higher Educational Science and Technology Program of Jinan City under Grant 2020GXRC057, and the State Scholarship Fund of the China Scholarship Council.

**Data Availability Statement:** The data presented in this study are openly available at <https://GAIA.didichuxing.com> (accessed on 13 March 2021).

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

## References

1. Mori, U.; Mendiburu, A. A review of travel time estimation and forecasting for advanced traveller information systems. *Transp. A Transp. Sci.* **2015**, *11*, 119–157. [\[CrossRef\]](#)
2. Chen, L.-W.; Hu, T.-Y. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873.
3. Sanger, T.D. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Netw.* **1989**, *2*, 459–473. [\[CrossRef\]](#)
4. Huang, W.; Song, G. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2191–2201. [\[CrossRef\]](#)
5. Li, Y.; Yu, R. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
6. Ma, X.-L.; Tao, Z.-M. 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. [\[CrossRef\]](#)
7. Wang, W.; Zhou, C. Cellular traffic load prediction with LSTM and Gaussian process regression. In Proceedings of the 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020.
8. Jiang, W.; Zhang, L. Geospatial data to images: A deep-learning framework for traffic forecasting. *Tsinghua Sci. Technol.* **2019**, *24*, 52–64. [\[CrossRef\]](#)
9. Han, L.; Zheng, K.; Zhao, L. Short-term traffic prediction based on deep cluster in large-scale road networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 12301–12313. [\[CrossRef\]](#)
10. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational inductive biases, deep learning, and graph networks. *arXiv* **2018**, arXiv:1806.01261.
11. Scarselli, F.; Gori, M. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [\[CrossRef\]](#)
12. Wu, Z.-H.; Pan, S.-R. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Zhang, J.-W. Graph neural networks for small graph and Giant network representation learning: An overview. *arXiv* **2019**, arXiv:1908.00187.
14. Guo, K. Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 1138–1149. [\[CrossRef\]](#)
15. Cirstea, R.G.; Guo, C.-J. Graph attention recurrent neural networks for correlated time series forecasting. *arXiv* **2021**, arXiv:2103.10760.
16. Cui, Z.; Henrickson, K.; Ke, R. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4883–4894. [\[CrossRef\]](#)
17. Ke, J.-T.; Feng, S.-Y. Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. *Transp. Res. Part C Emerg. Technol.* **2021**, *127*, 103063. [\[CrossRef\]](#)
18. Mohanty, S.; Pozdnukhov, A. Region-wide congestion prediction and control using deep learning. *Transp. Res. Part C Emerg. Technol.* **2020**, *116*, 102624. [\[CrossRef\]](#)
19. Leong, H.J.W. The distribution and trend of free speeds on two-lane rural highways in New South Wales. *Aust. Road Res. Board Conf.* **1968**, *4*, 791–814.
20. Katti, B.K.; Raghavachari, S. Modelling of mixed traffic speed data as inputs for the traffic simulation models. *Highw. Res. Bull.* **1986**, *28*, 35–48.
21. Kumar, V.M.; Rao, S.K. Headway and speed studies on two-lane highways. *Indian Highw.* **1998**, *26*, 23–26.
22. Dreyfus, S.E. Dynamic programming and the calculus of variations. *J. Math. Anal. Appl.* **1960**, *1*, 228–239. [\[CrossRef\]](#)
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
24. Luong, M.-T.; Pham, H. Effective approaches to attention-based neural machine translation. *arXiv* **2015**, arXiv:1508.04025.
25. Bahdanau, D.; Cho, K. Neural machine translation by jointly learning to align and translate. *arXiv* **2016**, arXiv:1409.0473.
26. Hu, J.; Shen, L. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [\[CrossRef\]](#)
27. Sun, Y.; Fisher, R. Object-based visual attention for computer vision. *Artif. Intell.* **2003**, *146*, 77–123. [\[CrossRef\]](#)
28. Mnih, V.; Heess, N. Recurrent models of visual attention. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27, pp. 2204–2212.
29. Lin, Z.; Li, M. Self-attention convlstm for spatiotemporal prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11531–11538.
30. Zhou, H. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 2–9 February 2021; Volume 35, pp. 11106–11115.
31. Cui, Z.; Ke, R.; Pu, Z.; Wang, Y. Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv* **2018**, arXiv:1801.02143.
32. Agarap, A.F. A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. *arXiv* **2019**, arXiv:1709.03082.

- 
33. Seo, Y.; Defferrard, M.; Vandergheynst, P.; Bresson, X. Structured sequence modeling with graph convolutional recurrent networks. In Proceedings of the International Conference on Neural Information Processing, Siem Reap, Cambodia, 13–16 December 2018; pp. 362–373.
  34. Yu, B.; Yin, H.-T. Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
  35. Wu, Z.-H.; Pan, S.-R. Graph waveNet for deep spatial-temporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.