A Spatial-Temporal Feature-Fusion Model Based on Graph Convolution Network for Traffic Flow Forecasting

Hao Shen and Min He

Abstract. With the rapid development of urban traffic, traffic forecasting has been attracting widespread attention. Accurate traffic forecaster can bring a convenient travel experience for the drivers and help the officers to direct traffic. Based on graph convolutional network, we propose a spatiotemporal feature fusion model named STFF-GCN to extract more valuable features from temporal segments by considering different periodic and spatial dependencies. Meanwhile, we use parallel dilated causal convolutions as residual connections to capture the relationship from long-term time series more easily. Experiments on the datasets of PEMS show that STFF-GCN outperforms other baseline models, which illustrates the effectiveness of our model.

Keywords. Spatial-temporal attention, traffic flow forecasting, graph convolutional network

1. Introduction

In recent years, with rapid socioeconomic development, the limited urban road resources cannot satisfy the increasing amounts of automobiles. Traffic congestion has become a considerable problem that affects human activities. Fortunately, great progress has been made in artificial intelligence (AI) field, which gives human life conveniences. At the same time, managing road traffic with AI technology has attracted much attention, and it is known as the intelligent transportation system (ITS). Reference [1] explored the prospects of intelligent transportation research and some new technologies. Traffic prediction is an essential part of intelligent transportation system, which aims to predict the future traffic flow of a given road according to the historical traffic data and the sensors deployed on the roads. Accurate traffic prediction can evaluate the traffic trend so that it helps to change and guide traffic flow distribution in advance to avoid congestion and traffic accidents. Effective prediction can also mitigate the crash sequences[2].

Traffic flow forecasting is essentially a time-series prediction. As we know, traffic data can constantly change over time and space, which means the spatiotemporal dependencies are complex and variable. Thus, traffic flow forecasting inevitably faces some challenges. Moreover, these spatiotemporal sequences are also influenced by external factors, such as weather conditions, accidents or other critical situations.

Although traffic datasets are usually generated from real traffic information, some abnormal and messy data are probably stored. These outliers often seriously impact on
the consistency of prediction results and the judgment of future trends that most of the previous research ignored. Furthermore, most methods are mainly focused on extracting the overall temporal and spatial characteristics while neglecting some local information. Even though some studies have considered identifying temporal dependency and spatial dependency respectively, it is still hard to completely detach each dependency from the extracted features. Additionally, when the layers of GCN are increased, the network's parameters cannot get effectively trained. Therefore, we propose a GCN-based spatiotemporal feature fusion model called STFF-GCN to overcome the above problems, and our main contributions are as follows:

- We introduce a filter before training to delete the outliers that previous studies rarely considered, which greatly contributes to improving the quality of training data.
- Obviously, weekly and daily data are more periodic than recent data, and the feature propagation of recent data depends much more on the information transmitted through nodes. Thus, we adopt different attention mechanisms to extract more valuable features of temporal segments. And then, we designed a new splicing network to aggregate these temporal features with spatial features to mine the correlation features.
- We design parallel dilated causal convolutions as residual connections to enable the network to have a substantial receptive field, which could easily capture relationships from long-term time series.

2. Related Work

Traffic flow is the number of vehicles passing through a spatial unit, such as a road segment or a traffic sensor, in a given time. An accurate prediction benefits various applications, e.g., traffic congestion control, traffic light control, and vehicular cloud[3].

Traffic flow forecasting has attracted increasing attention in recent years. Researchers have been exploring how to precisely predict short-term traffic conditions with the real-time traffic information, which contributes to implementing traffic control and guidance. Reference [4] summarized the researches of traffic forecasting. Some classical machine learning models, such as Support Vector Regression (SVR)[5], were adopted to obtain features by training the regression model. Authors introduce a novel machine learning architecture for an efficient estimation of the probabilistic space-time representation of complex traffic scenarios in [6].

With the development of deep learning technology, deep learning has been introduced to predict traffic flow. Due to the perfect performance of CNN in image processing, CNN was first involved in extracting the spatial correlation from traffic data. However, CNN-based methods do not play well for graph-based road networks. Recurrent neural network can model temporal correlations due to its short-term memory. As an improved recurrent neural network, LSTM releases the problem of long-distance dependencies that RNN cannot handle. DCRNN[7] uses bidirectional random walk to capture spatial dependencies and uses an encoder-decoder with predetermined samples to capture temporal dependencies. However, the above methods failed to fully explore the global and local correlations from traffic data. Unlike the clear network structures such as images and audio, traffic data is classic non-Euclidean-structured graph data. Early methods only considered the dynamic changes of traffic conditions with time,
while ignoring the interdependence between available spaces. Therefore, GCN is involved to the traffic forecasting field due to its ability of directly processing graph-structured data. To comprehensively describe the trend of combining graph data with deep learning technology, reference [8] investigated various graph-based deep learning architectures that are used in many traffic scenarios. Some surveys, such as [9] and [10], studied traffic prediction methods based on deep learning from multiple perspectives and identified future research directions. For example, T-GCN[11] combines graph convolutional networks and gate recurrent units to capture the dynamic changes of road network topology and road traffic data, respectively. STGCN[12] proposed a framework composed of spatial-temporal blocks in which the training is faster and the parameters are not so many due to its pure convolutional operation.

Some studies such as GAT[13] incorporate attention mechanism into the propagation step and utilize multi-head attention mechanism to stabilize the learning process. The spatiotemporal synchronization modeling mechanism of STSGCN[14] can effectively capture the complex local spatiotemporal correlation. ASTGCN[15] uses spatiotemporal attention to capture dynamic spatiotemporal correlations on transportation network. They have intercepted three segments with different time steps along the time axis, and the time steps are recent, daily, and weekly components.

Nevertheless, extracting the features from recent, daily-period, and weekly-period components by merely weighted sum is not reasonable. Although period correlation does exist in daily-period and weekly-period components, the periodicity of recent component should also be involved. And its feature expression dependents more on the latest traffic flow propagated through nodes.

3. Methodology

To resolve the above problems, we propose a spatiotemporal feature-fusion framework based on graph convolutional network called STFF-GCN to mine the variations of different features in time and space. The framework is shown as Fig.1, which includes data preprocessing, self-attention module, spatiotemporal feature fusion, graph convolution module and residual module composed of multi-layer WaveNet. It can enhance the predictability of original time series by capturing temporal dependencies on multiple time steps.

$X_h$, $X_d$, and $X_w$ respectively represent the recent segment and the daily segment, and the weekly segment after data preprocessing. Then, after giving weight through temporal attention and spatial attention, we can get $X_{wh}$ from $X_w$ and $X_h$ with weekly and spatial feature fusion, and get $X_{dh}$ from $X_d$ and $X_h$ with daily and spatial feature fusion.
3.1. Preliminaries

Definition 1. (Traffic Graph): The transportation network can be described as $G = (V, E, A)$, where $V$ is the node set, $E$ is the edge set, and $A$ is the adjacency matrix.

Definition 2. (Graph signal tensor): After excluding some uncontrollable external factors, the transportation network can be described as follows: $x_i^t \in \mathbb{R}^F$ is the observation of node $i$ at time $t$, $F$ represents the length of the observation vector. $X_i = (x_i^1, x_i^2, \ldots, x_i^N) \in \mathbb{R}^{N \times F}$ represents the state of $N$ nodes at time $t$. And $\mathcal{X} = (X_1, X_2, \ldots, X_T) \in \mathbb{R}^{T \times N \times F}$ represents the state of all the nodes at all the time steps.
The prediction function is defined as \( y = f(X; G) \), where \( y \) is the traffic state we want to predict.

### 3.2. Data Preprocessing

The data preprocessing module contains two operations:

(i). When transmitting the collected information, environmental interference or human factors may cause some errors or data loss. These data are called outliers that can reduce the accuracy of prediction. Therefore, we introduce Hampel filter to remove or recover these outliers to eliminate or minimize the negative influence.

(ii). In order to extract spatiotemporal features respectively, we apply a data processing method similar to ASTGCN after dealing with the outliers. As shown in Fig.1, the data processed through data preprocess module are divided into recent segment, daily segment and weekly segment, which are presented as \( X_k \), \( X_d \) and \( X_w \) respectively.

Hampel filter is a kind of decision filter that replaces the data window's central value with the median, in which any point far enough from the median is regarded as an outlier [16]. Hampel filter is defined as follow:

\[
y_k = \begin{cases} 
  x_k & \text{if } |x_k - m_k| \leq tS_k, \\
  m_k & \text{otherwise,}
\end{cases}
\]  

The data window’s central point \( x_k \) is deemed an outlier if it is larger than \( t \) times the estimated value of median absolute deviation (MAD) and it will be replaced by the median. Generally, the default value of \( t \) is 3. Here, \( m_k \) is the median of \( x_k \) and \( S_k \) is the MAD scale estimate, defined as:

\[
S_k = k \times \text{median}_{j=1\cdots K}(|x_{k-j} - m_k|),
\]  

\[
k = \frac{1}{\sqrt{2 \ erfc^{-1}(1/2)}} \approx 1.4826,
\]  

As shown as Fig.2, it is easy to understand that period correlation does exist in day-period and weekly-period components. However, for the recent component, its feature expression is also related to those nodes that the recent traffic flow propagates through. The visualization of a traffic flow from July 11 to July 19, 2016, and the data come from a real dataset. It can be seen that daily-period and weekly-period have apparent periodicity.
Assume \( q \) is sampling frequency, \( t_0 \) is current time, and the size of prediction window is \( T_p \). Along the time axis, traffic flow can be split into time series in hour-long segments presented as \( T_h \). In the same way, we can divide the traffic flow into daily segments and weekly segments defined as \( T_d \) and \( T_w \), respectively. Thus, the recent segment \( X_h \), the daily-periodic segment \( X_d \) and the weekly-periodic segment \( X_w \) are expressed as:

\[
X_h = (X_{h, t_0 + 1}, X_{h, t_0 + 2}, \ldots, X_{h, t_0 + T_h}) \in \mathbb{R}^{N \times F \times T_h},
\]

\[
X_d = (X_{d, (t_0/T_d)q + 1}, X_{d, (t_0/T_d)q + 2}, \ldots, X_{d, (t_0/T_d)q + T_d}) \in \mathbb{R}^{N \times F \times T_d},
\]

\[
X_w = (X_{w, (t_0/T_w)q + 1}, X_{w, (t_0/T_w)q + 2}, \ldots, X_{w, (t_0/T_w)q + T_w}) \in \mathbb{R}^{N \times F \times T_w},
\]

where \( T_h \), \( T_d \) and \( T_w \) are multiples of \( T_p \).

### 3.3. Attention Mechanism

In our model, we use temporal and spatial attention to capture the correlation between time and space for different time-series segments according to the specificity of spatiotemporal characteristics.

- **Spatial attention:** Regarding the nodes, their traffic conditions are highly correlated with spatial dimension in a short time, the features of spatial nodes in recent data are more useful. Therefore, we adopt spatial attention mechanism to deal with recent segments, and the formula is given by:
where $\sigma$ denotes the sigmoid function, and $f^{T \times 7}$ represents a convolution operation with filter size of $7 \times 7$. $X^{r-1} = (X_1, X_2, ..., X_{T-r}) \in \mathbb{R}^{F \times T \times r}$ is the input of the $r^{th}$ spatial block. $F_{r-1}$ is the number of input data's features in the $r^{th}$ layer. is the length of temporal dimension in the $r^{th}$ layer.

- Temporal attention: Obviously, daily and weekly components have cyclical correlation on time dimension, hence, we use channel attention to set different weights for the data. The temporal correlation matrix is given by:

$$E = \sigma(\text{MLP}(\text{AvgPool}(X^{r-1}))) + \text{MLP}(\text{MaxPool}(X^{r-1})))$$

$$= \sigma(\mathbf{W}_l (\mathbf{X}_0 \mathbf{X}_l^{r-1})_{\text{avg}} + \mathbf{W}_l (\mathbf{X}_0 \mathbf{X}_l^{r-1})_{\text{max}}),$$

where $\sigma$ denote the sigmoid function, $\mathbf{W}_0 \in \mathbb{R}^{T \times 1 \times T - 1}$ and $\mathbf{W}_l \in \mathbb{R}^{T \times 1 \times T - 1}$ are learnable. When we take the temporal correlation matrix $E$ as the corresponding output, and obtain $X^{r-1} = (\tilde{X}_1, \tilde{X}_2, ..., \tilde{X}_{T-r}) = (X_1, X_2, ..., X_{T-r})E \in \mathbb{R}^{N \times F \times T \times r}$ to extract the periodic features by combining relevant weighted information.

### 3.4. Graph Convolutional Network Layer

In GCN, the aggregation process for node information can be expressed as:

$$X_{s} = \sigma \left( \sum_{k=0}^{K} \theta_k T_k (\tilde{L}) X \right),$$

where $\sigma$ is a nonlinear activation function and ReLU(). In order to incorporate the dynamical attributes of the nodes, we aggregate the information from the graph signal $\mathcal{X} \in \mathbb{R}^{T \times N \times F}$ at each time step by using the $K$-th order Chebyshev polynomial $T_k$ and $K$ is set as 3 in this paper, which means $k$ ranges from 0 to $K$. And $\mathcal{X}_{s} \in \mathbb{R}^{T \times N \times F}$ is the aggregated information from graph $\mathcal{X}$. $\theta_k$ is the weight parameter. For Chebyshev polynomial, the scaled Laplacian matrix is given by $\tilde{L} = 2L/\lambda_{\text{max}} - I_n$, where the normalized Laplacian matrix $L$ is defined as $L = I_n - D^{-1/2} \tilde{A} D^{-1/2}$. $\lambda_{\text{max}}$ is the maximum eigenvalue of matrix $L$, $I_n$ is an identity matrix, and $A \in \mathbb{R}^{N \times N}$ denotes the adjacent matrix which is weighted by a vertex distance function. $D$ is the degree matrix, which is defined as $D_{ij} = \sum_j A_{ij}$. 
3.5. Residual Connection

In order to reduce the complexity and over-fitting and to avoid gradient vanishing, we introduce a multi-layer WaveNet[17] as residual connections, which provides dilated casual convolutions and gating mechanism.

![Overview of Dilated Causal Convolution Architecture and Gating Mechanisms.](Image)

Stacked dilated causal convolution enables the network to have a substantial receptive field while preserving the resolution of input data and computational efficiency. Empirically, gated activation is more robust than ReLU for audio data, and it also performs well to time-series data. However, ReLU may cause the processed data to be sparse. Therefore, we choose gated activation mechanism to make the information smoother when passing through the multi-layer WaveNet network.

4. Datasets and Experiment

We testify the efficiency of our model on four real-world datasets, i.e., PeMS03, PeMS04, PeMS07 and PeMS08. These traffic data were provided by Caltrans Performance Measurement System (PeMS) which collects real-time information every 30 seconds from more than 39,000 detectors deployed on the highways that across all major metro areas of California. Traffic sensors deployed on the road network present the nodes and edges mean the connections between two nodes. The details are listed in Table 1. In fairness, we adopt the same evaluation metrics to validate different methods.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Samples</th>
<th>Nodes</th>
<th>Edges</th>
<th>Time Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>PeMS03</td>
<td>26208</td>
<td>358</td>
<td>547</td>
<td>09/01/2018-11/30/2018</td>
</tr>
<tr>
<td>PeMS04</td>
<td>16992</td>
<td>307</td>
<td>340</td>
<td>01/01/2018-02/28/2018</td>
</tr>
<tr>
<td>PeMS07</td>
<td>28224</td>
<td>883</td>
<td>866</td>
<td>05/01/2017-08/31/2017</td>
</tr>
<tr>
<td>PeMS08</td>
<td>17856</td>
<td>170</td>
<td>295</td>
<td>07/01/2016-08/31/2016</td>
</tr>
</tbody>
</table>

At the same time, we take three metrics to evaluate the performance, which are MAE (mean absolute error), RMSE (root mean square error) and MAPE (mean absolute
percentage error. They are given by: $MAE(\hat{y}_i, y_i) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$ ,

$RMSE(\hat{y}_i, y_i) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$ and $MAPE(\hat{y}_i, y_i) = \frac{1}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{y_i} \times 100\%$ .

Obviously, the lower the values, the better the performance.

4.1. Experimental Settings and Results

We also conduct experiments to compare the performance of STFF-GCN with some classic baseline models, such as SVR, STSGCN and so on. Similar to the setting in STSGCN, all data from the four datasets are split into training, validation, and test sets in a ratio of 6:2:2. The normalized data are fed into the model and optimized by automatic differentiation and Adam.

The prediction results in the next 60 minutes are listed in Table 2, from which we can observe that STFF-GCN performs best compared to other baselines. Specifically, deep learning methods perform better than traditional statistical methods, for traditional methods such as SVR only considered temporal correlations spatial dependencies. In contrast, deep learning models utilize the spatial-temporal information, which will be of benefit to performance improvement. Except for STSGCN, other baseline methods based on deep learning utilize two modules to model spatial dependencies and temporal correlations separately, which neglects the complex interaction between spatial information and temporal information. Hence, STSGCN outperforms other models. However, STSGCN is focused on the local spatial-temporal correlations while regarding turtles as global dependencies.

Table 2. Performance comparison of different methods on PEMS03, PEMS04, PEMS07 and PEMS08.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>SVR</th>
<th>STGCN</th>
<th>DCRNN</th>
<th>Graph WaveNet</th>
<th>STSGCN</th>
<th>OURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEMS03</td>
<td>MAE</td>
<td>22.03</td>
<td>17.68</td>
<td>17.99</td>
<td>19.12</td>
<td>17.48</td>
<td>16.24</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>35.31</td>
<td>30.51</td>
<td>30.31</td>
<td>32.77</td>
<td>29.21</td>
<td>28.52</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>23.24</td>
<td>17.49</td>
<td>18.34</td>
<td>18.89</td>
<td>16.78</td>
<td>15.23</td>
</tr>
<tr>
<td>PEMS04</td>
<td>MAE</td>
<td>28.65</td>
<td>22.43</td>
<td>21.22</td>
<td>24.89</td>
<td>21.19</td>
<td>18.63</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>44.57</td>
<td>35.20</td>
<td>33.44</td>
<td>39.66</td>
<td>33.65</td>
<td>30.64</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>50.19</td>
<td>40.55</td>
<td>38.61</td>
<td>41.50</td>
<td>39.03</td>
<td>32.99</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>15.61</td>
<td>11.71</td>
<td>11.82</td>
<td>11.97</td>
<td>10.21</td>
<td>8.19</td>
</tr>
<tr>
<td>PEMS08</td>
<td>MAE</td>
<td>23.25</td>
<td>17.86</td>
<td>16.82</td>
<td>18.28</td>
<td>17.13</td>
<td>13.91</td>
</tr>
<tr>
<td></td>
<td>RMSE</td>
<td>36.12</td>
<td>27.91</td>
<td>26.36</td>
<td>30.05</td>
<td>26.80</td>
<td>22.99</td>
</tr>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>14.66</td>
<td>11.24</td>
<td>10.92</td>
<td>12.15</td>
<td>10.96</td>
<td>9.09</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, we design a GCN-based model called STFF-GCN to fuse the spatiotemporal dependencies. STFF-GCN can effectively capture the complex spatiotemporal dependencies of nodes with different timestamps. Experiments on PEMS
datasets show that RMSE, MAE and MAPE of our model are better than other baseline models such as STSGCN.

In the future, we take into account more abnormal conditions such as traffic congestion to optimize the network structure so that it can better extract the spatial features and contribute to long-term prediction.

References


