Received: 24 March 2024 | Revised: 13 June 2024 | Accepted: 17 July 2024 | Published online: 31 July 2024

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2024, Vol. 3(4) 395–403
DOI: 10.47852/bonviewJCCE42022914

BON VIEW PUBLISHING

Multi-view Traffic Flow Prediction Model Based on Spatiotemporal Convolution

Xiaoyan Zhu[1,*]

[1] *College of Computer Science and Technology, Qingdao University, China*

Abstract: Predicting traffic flow has always been a significant task in intelligent transportation systems. Due to the substantial temporal and spatial dependencies of traffic flow sequences, accurately predicting traffic flow poses a considerable challenge. Many existing works primarily rely on recurrent neural networks, graph neural networks, and Transformer models to establish traffic flow prediction models. To better extract features and enhance efficiency, a traffic flow prediction model based on multi-view spatiotemporal convolution (MVSC) is proposed. This model learns the representation of sequence data at the input encoding layer and incorporates location and time information. In the spatiotemporal feature representation learning layer, considering the diverse periodic patterns in sequences, several representation learning modules are designed, conducting local spatiotemporal feature exploration through one-dimensional convolution and then accomplishing global spatiotemporal feature mining based on causal convolution. To further enhance the model's utilization of spatiotemporal features, a channel attention mechanism is introduced at the prediction layer. The forecasting method employed in the study is direct multi-step, and subsequent experiments conducted on two real datasets demonstrate that the MVSC model exhibits a certain degree of superiority in MAE, RMSE, and MAPE for both short-term and long-term predictions compared to existing models. And through the latest experiments and investigations, it has been found that MVSC has improved MAPE performance by about 1.2% compared to recent models such as RTGCN and STRGCN, achieving the intended outcomes.

Keywords: traffic engineering, traffic flow prediction, spatiotemporal convolution, attention mechanism, Transformer

1. Introduction

Time series prediction technology is one of the indispensable human intelligence technologies in the field of transportation [1, 2]. For example, traffic management departments can optimize traffic signal control and alleviate congestion based on vehicle flow prediction at each intersection [3]. Bus companies can reasonably plan departure frequency based on road flow to reduce citizens' waiting time [4–6]. Emergency rescue department, in the face of fire, dangerous chemical explosion, and other emergencies, the accurate prediction of traffic flow at each intersection to help emergency traffic route planning is of great significance. The state also clearly proposed in the "14th Five-Year Plan" to strengthen the deep application of big data and artificial intelligence technology in the transportation industry.

The current mainstream deep learning-based traffic flow prediction models can be summarized into the following two categories [7]. (1) Model based on recurrent neural network (RNN). For instance, Huang et al. [5] devised a segmented attention module based on GRU network to achieve traffic flow prediction [8]. Then, the ConvLSTM model, based on LSTM, combined with CNN to learn temporal features of sequences and achieve prediction. In last several years [9], graph neural networks (GNN) have been widely applied in feature learning modeling of social networks, traffic networks, molecular structures, and other data due to their powerful spatial network modeling capabilities [10, 11]. Therefore, scholars will also incorporate GNN in RNNs. For example, Yu et al. [12] used graph network to model the spatial relationship of traffic at each intersection in the traffic network and used LSTM to learn the temporal characteristics of the spatial relationship [13]. (2) Models based on Transformer framework. For example, Xu et al. [14] build a time-sequence representation module and a spatial representation module based on Transformer [15], making traffic flow prediction a reality. And many Transformer variants have been proposed during these past few years, including informer [12], autoformer [16], FEDformer [17], etc., which can be used for AC flow premeasurement. The GNN network can also be combined with the Transformer class model [10, 11]. For example, Wang et al. [18] constructed a spatial-temporal feature extraction module based on GCN and Transformer through multi-scale partitioning of sequences [19, 20].

The Transformer model operates through multi-head self-attention, effectively capturing interdependencies among elements within sequential data [14]. Nonetheless, despite the non-sequential nature of the self-attention mechanism, it excels at capturing contextual semantics in textual data without issue [21]. For time series data such as traffic flow, ignoring the sequence of data is not conducive to its feature mining. Therefore, literature focuses on the feasibility of Transformer for sequence prediction. In addition, this paper argues that the Transformer model introduces many redundant operations in the attention module (Q, K, V calculation part) [22], and this increases the computational complexity of the model to a certain extent. Thus, although the Transformer class model has

*__Corresponding author:__ Xiaoyan Zhu, College of Computer Science and Technology, Qingdao University, China. Email: zhuxiaoyan@ubinet.cn

a good performance in sequence prediction [15], this paper does not build a traffic flow prediction model relying on Transformer framework, but a novel model based on convolutional operations. On the one hand, the volume operation is more efficient than the Transformer model in terms of computational efficiency [15]. On the other hand, convolutional-based models have very powerful feature mining capabilities for image data, and there are also some models that mine features of time series data based on one-dimensional convolution design models, such as TCN [23]. For time series data such as traffic flow, the local fluctuation is very large, and the whole has a certain periodicity. Based on the convolutional design prediction model, this paper considers that the following problems need to be solved: (1) Circular networks naturally have the ability of timing mining, while convolutional networks do not have such ability, so how to learn the timing relationship of traffic flow data is the first problem to be considered in model design; (2) Traffic flow data have a certain periodicity, and the local fluctuation is large, so how to effectively mine the cycle pattern is also an important issue to be considered.

In this study, a multi-view spatiotemporal convolution (MVSC) model was proposed for traffic flow prediction. The primary contributions of the MVSC model are outlined as follows: (1) The learning layer of spatiotemporal feature representation of traffic flow data based on convolutional network is designed, and multi-view mode is introduced to complete spatiotemporal feature learning of different cycle modes with different views; (2) A prediction layer relies on channel attention mechanism that was designed for multi-dimensional time series data prediction. Channel attention completed the filtering of multi-view spatiotemporal feature information and provided effective spatiotemporal feature information for the prediction module. Experiments were conducted on two traffic datasets, and the experimental results verified the effectiveness of the model.

2. Relevant Theoretical Knowledge

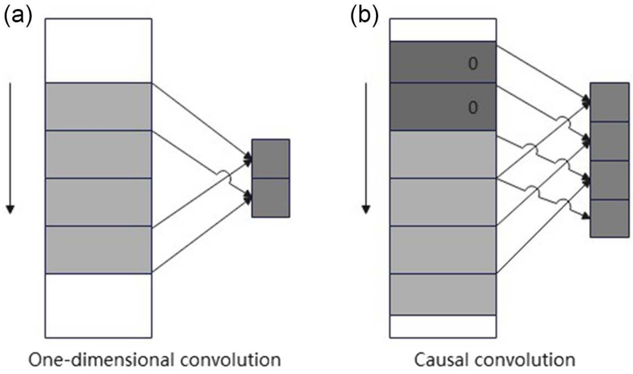
2.1. Convolutional network

Convolutional network realizes feature extraction of data based on convolutional kernel. For image data, two-dimensional convolutional networks are commonly used. For time series data, two typical 1D convolutional networks are introduced in this paper. The characteristic of a two-dimensional volume network is that the convolution kernel moves in two dimensions.

The working principle of a conventional one-dimensional convolutional network is shown in Figure 1(a), where the time series dimension is 1, the quantity of input and output channels is 1, and the size of the convolutional kernel is 3. The convolution kernel operates from left to right and performs a dot product operation on an input subsequence of equal length to obtain the corresponding output. If it is desired for the input sequence to match the length of the output sequence, padding can be applied to the input sequence. When the output channel is greater than 1, repeat the above process for each output channel with different kernel matrices and then stack the output vectors sequentially.

The working principle of causal convolution is shown in Figure 1(b), which is characterized by the fact that an element in the output sequence can only depend on the element before it is in the input sequence. To ensure that the length of the output tensor and input tensor is the same, zero padding is applied after each convolution. When the output channel is greater than 1, the operation method is the same as that of a one-dimensional convolutional network.

Figure 1
Convolutional example (a) One-dimensional convolution and (b) Causal convolution



2.2. Attention mechanism

The function of attention mechanism is to train the model to assign different attention weights to input data [24]. In addition, it can make the model only focus on a part of the input sequence, thereby improving the efficiency and performance of the model without falling into the problem of insufficient resources [25]. The attention mechanism is expressed in a more general way [22]:

$$\text{Attention}\,(Q, K, V) = \sum_{j=1}^{n} \text{Softmax}\,\left(\text{Similarity}\left(\mathrm{Q}, \mathrm{k_j}\right)\right) * \mathrm{V_j} \tag{1}$$

where Q is the query matrix; K is the keyword matrix; V is the value matrix.

Multi-head attention of a Transformer can be expressed as:

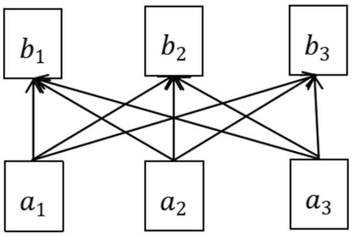
$$\mathrm{H} = \text{concat}\ (\mathrm{b_1}, ...\mathrm{b_u})\ \mathrm{W^o} \tag{2}$$

The computational process is as below: firstly, u attention representations are concatenated, and then matrix multiplication is performed with $\mathrm{W^o}$. Each header is implemented based on a single self-explanatory mechanism, i.e., a function of Q, K, V [22]:

$$b_i = \text{softmax}\left(\frac{\mathrm{QK^T}}{\sqrt{\mathrm{d_k}}}\mathrm{V}\right) \tag{3}$$

Figure 2 shows a schematic of the self-attention mechanism. It can be seen that the self-attention mechanism considers all the input vectors, but ignores the position information of the vectors. And a1, a2, and a3 represent the vectors used to compute the attention weights for queries (Q), keys (K), and values (V), b1, b2, b3 represent other parameters related to attention calculation.

Figure 2
Examples of self-attention mechanisms



3. MVSC Model

3.1. Problem definition

The traffic flow prediction problem is to predict the data $Y \in R^{T \times N}$ of all observation points in the future T period by learning function F based on the given historical observation data $X \in R^{H \times N}$, where N is the number of observation points and H is the window size of the historical sequence input [26]. The formula is defined as below:

$$Y = F(X) \tag{4}$$

3.2. Model overview

Figure 3 shows the overall architecture of a MVSC model. The model contains three parts, which are input coding layer, spatiotemporal feature representation learning layer, and prediction layer. First, the input coding layer encodes the input sequence and adds time information and position information to increase the input characteristics of each observation point. Secondly, there are multiple spatiotemporal convolution modules in the spatiotemporal feature representation learning layer, and the convolution kernel used in each module is different in size. In each module, one-dimensional convolution, causal convolution, and one-dimensional deconvolution are used to complete spatiotemporal feature extraction of sequences [19]. Finally, in the prediction layer, traffic flow prediction is completed based on the channel timing attention module, two-dimensional convolution, and multi-layer perceptron.

3.3. Enter the coding layer

At present, the mainstream Transformer variant models such as Informer [12], Autoformer [16], FEDformer [17], part of the input of the decoder of these models are the input vector of the encoder, and the other part of the information is constant. Inspired by this, this paper adopts a simpler zero-complement strategy in the input coding layer to keep the following convolution module working normally. In addition, this paper refers to the setup of the FEDformer model [17], using three parts of input:

$$X_{emb} = V_e + P_e + T_e \tag{5}$$

where $X_{emb} \in R^{(H+T) \times d}$; $V_e$is the representation vector of the input sequence supplemented by 0; P_e is the sequence of location information encoding; $T_e$is the encoding of sequence time information $X_{time} \in R^{(I+O) \times 5}$, which is mainly composed of month, day, week, hour and minute. The formula for calculating V_e, T_e, and P_e is as follows:

$$V_e = VE([X||X_0]) \tag{6}$$

$$P_e = \begin{cases} \sin\left(\frac{t}{10000^{\frac{2i}{d}}}\right) & \text{if} \quad t = 0, 2, 4, \ldots \\ \cos\left(\frac{t}{10000^{\frac{2i}{d}}}\right) & \text{if} \quad t = 1, 3, 5, \ldots \end{cases} \tag{7}$$

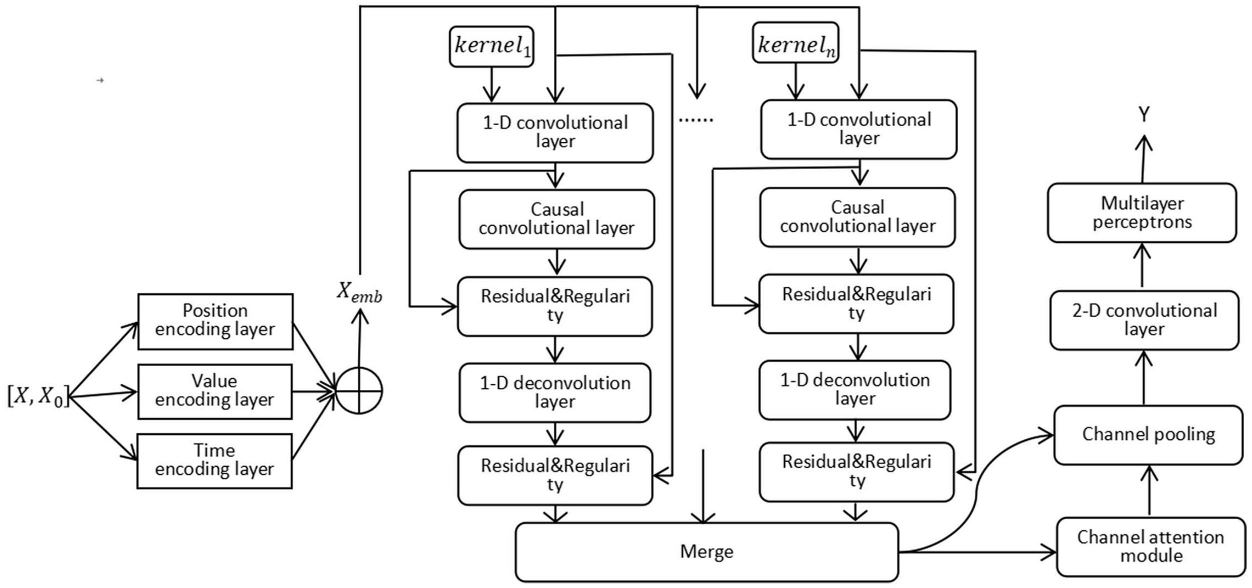
$$T_e = TE\ (X_{time}) \tag{8}$$

In the formula, VE is a value coding network, implemented using a one-dimensional convolutional network, with input channel N and output channel d; TE is a time-coded network consisting of 5 fully connected layers. Each fully connected layer encodes the month, day, week, hour, and minute respectively, and then sums the feature vectors. The input dimension of each fully connected layer is 1, and the output dimension is d. In formula (7), t corresponds to the actual position of the input value in the sequence.

3.4. Spatiotemporal features represent the learning layer

The spatiotemporal feature representation learning layer is composed of multiple views, each view map corresponds to different spatiotemporal feature modules, and convolution kernel of different sizes is used for different views in feature mining. The volume kernel size corresponding to each view is kernel = k, k ∈{H/4,

Figure 3
Overview of the MVSC model



H/6, H/8, H/16}, in this paper, H is set to 96, taking 4 views, then the convolution kernel of each view is 24,16,12,8, respectively. The following details the implementation steps of spatiotemporal feature mining in each view.

1) Local spatiotemporal feature mining

X_{emb} contains historical traffic flow information and time information for all traffic intersections. It should be noted that X_{emb}has done preliminary representation learning for the spatial correlation of each intersection, that is, the dimension of input sequence data X is changed from $R^{H\times N}$ to $R^{(H+T)\times d}$ by inputting into the coding layer. In this paper, the spatiotemporal characteristics of a periodic model are further extracted based on one-dimensional convolutional networks. Define the periodic pattern according to the size of the convolution kernel, and then, the convolution operation is as follows:

$$X^{k}_{local} = \text{Convld}\left(\text{Padding}\left(X_{emb}\right)\right)_{kernal = k} \tag{9}$$

The convolution kernel size of one-dimensional convolution is set to $k \in \{H/4, H/6, H/8, H/16\}$,and the moving step is stride = k, that is, the convolution is used for feature extraction every cycle. After passing through the convolutional network, the eigenvector is compressed, and the dimension of X^{k}_{local} is $R^{\frac{H+T}{k}\times d}$.

2) Global spatiotemporal feature mining

X^{k}_{local} only contains spatiotemporal features of local cycles. In order to explore spatiotemporal features across the entire input time scale, this paper designs a causal convolution module:

$$X^{k}_{causal} = \text{CasualConv}\left(X^{K}_{local}\right) \tag{10}$$

In order to ensure that the X^{k}_{causal} and X^{k}_{local} have the same dimension, padding is required for X^{k}_{local}. After the causal convolution operation, the activation layer and Dropout operations are programmed. The activation function adopts Tanh, Dropout value is 0.1, that is:

$$X^{k}_{causal} = \text{Dropout}\left(\text{Tanh}\left(X^{K}_{causal}\right)\right) \tag{11}$$

The role of X^{k}_{causal} is to aggregate information on different local spatiotemporal features to complete the mining of spatiotemporal features on the entire input time scale.

Inspired by the encoding and decoding framework, this paper expects to achieve prediction by decoding. Therefore, a one-dimensional deconvolution module is designed to extend the dimension of spatiotemporal feature vector, and the global spatiotemporal feature X^{k}_{global} is obtained. The execution steps are as follows:

$$X^{k}_{res} = \text{Normal}\left(X^{k}_{causal} + X^{k}_{local}\right) \tag{12}$$

$$X^{k}_{trans} = \text{ConvldTrans}(X^{k}_{res}) \tag{13}$$

$$X^{k}_{trans} = \text{DropOut}\left(\text{Tanh}\left(X^{k}_{trans}\right)\right) \tag{14}$$

$$X^{k}_{global} = \text{Norm}\ \left(X^{k}_{trans} + X_{emb}\right) \tag{15}$$

Before performing deconvolution, residual joins were introduced to fuse information from X^{k}_{causal} and X^{k}_{local}, as shown in formula (12). After the one-dimensional deconvolution operation, the activation layer and Dropout operation are designed. The activation function uses Tanh and the value of Dropout is 0.1. Finally, the global space-time feature $X^{k}_{global} \in R^{(H+T)\times d}$ is obtained by fusing the information of X^{k}_{trans} and X_{emb} through residual connection.

3.5. Prediction layer

In the prediction layer, Merge operation is used to merge the global spatiotemporal features in different view images. Due to the initiation of channel attention mechanism, this paper believes that the different dimensions of spatiotemporal characteristics after fusion have different importance to prediction, so the channel attention module is introduced.

First of all, Merge operation of multi-view global spatiotemporal features is implemented as follows:

$$X_{global} = \text{concat}\left(X^{1}_{global}, \ldots, X^{k}_{global}\right) \tag{16}$$

Then, the channel attention weight M_c is calculated based on the channel attention module, as shown in Figure 4. First, the maximum pooling and average pooling operations are performed on X_{global}, respectively, and the representation vectors $X^{max}_{global} \in R^{l\times d}$ and $X^{avg}_{global} \in R^{l\times d}$ are obtained. These two representation vectors pass through the multi-layer perception, respectively. And then merge the output vectors. The whole process is represented as [27]:

$$X^{max}_{global} = \text{MaxPool}\ \left(XV_{global}\right) \tag{17}$$

$$X^{avg}_{global} = \text{AvgPool}\left(X_{global}\right) \tag{18}$$

$$M_c = \sigma\left(\text{MLP}\left(X^{max}_{global}\right) + \text{MLP}(X^{max}_{global})\right) \tag{19}$$

X_{global} is then channeled as follows:

$$X_c = M_c \odot X_{global} \tag{20}$$

$$X_c = \text{concat}\left(\max\left(X_C\right),\ \text{avg}\ \left(X_c\right)\right) \tag{21}$$

The dimension of X_c after channel pooling is $R^{(H+T)\times 2}$. Finally, the prediction is realized through 2D convolution layer and multilayer perceptron:

$$\hat{Y} = \text{MLP}\left(\text{Conv2d}\ \left(X_c\right)\right) \tag{22}$$

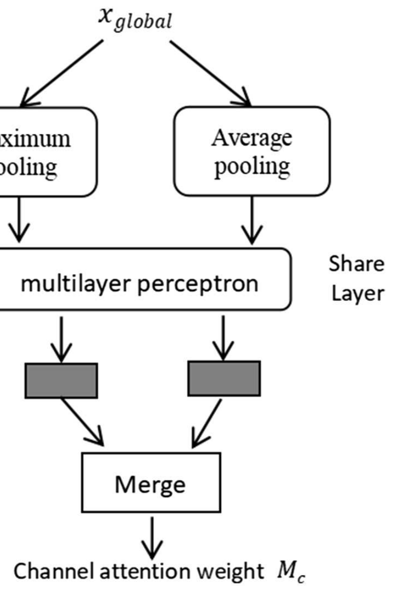
3.6. Training

In the training stage of the MVSC model, the error between the predicted traffic flow and the real traffic flow is used to guide the model to update parameters. The loss function is as follows:

$$\text{Loss} = \frac{1}{nT}\sum_{i}^{n}\sum_{j}^{T}(\hat{Y}_{i,j} - Y_{i,j})^2 \tag{23}$$

where $\hat{Y}_{i,j}$ is the predicted traffic flow; $Y_{i,j}$ is the real traffic flow; n represents the number of training samples; T is the predicted time step.

Figure 4
Channel attention module



4. Experiment

4.1. Data set

This paper verifies the performance of MVSC on two public transportation datasets.

1) METR-LA: Los Angeles Freeway traffic dataset, from March 1, 2012 to June 30, 2012, deployed 207 ring sensors collected on the freeway, sampling frequency of 5 min, sequence length of 34 272.
2) PEMS-BAY: BAY Area of California traffic dataset, from January 1, 2017 to May 31, 2017, collected by the Traffic Performance Measurement System (PEMS) (325 sensors), with a sampling frequency of 5 min and a sequence length of 52 116.

Each dataset is divided into training, testing, and validation sets in a 7:2:1 ratio. And the method used is cross-validation, and each sample is constructed with a sliding window with a window length of H+T and a sliding step size of 5. For the convenience of comparing model performance, the Z-score method was adopted to standardize the data by referring to the data preprocessing method of STSGCN model [28]. The concrete calculating method is as follows:

$$Z = \frac{\mathrm{X} - \overline{\mathrm{X}}}{\mathrm{std(X)}} \tag{24}$$

where $\overline{\mathrm{X}}$ is the mean value of the data set; std (X) is the variance.

4.2. Evaluation index and comparison model

In this article, mean absolute error (MAE), RMSE, and mean absolute percentage error are used. MAPE was used to evaluate model performance. The specific calculation formula is as follows:

$$\mathrm{MAE} = \frac{1}{\mathrm{nT}} \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \sum\nolimits_{\mathrm{j}}^{\mathrm{T}} \left| \hat{\mathrm{Y}}_{\mathrm{i,j}} - _{\mathrm{i,j}} \right| \tag{25}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{nT}} \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \sum\nolimits_{\mathrm{j}}^{\mathrm{T}} \left(\hat{\mathrm{Y}}_{\mathrm{i,j}} - \mathrm{Y}_{\mathrm{i.j}} \right)^2} \tag{26}$$

$$\mathrm{MAPE} = \frac{100\%}{\mathrm{nT}} \sum\nolimits_{\mathrm{i}=1}^{\mathrm{n}} \sum\nolimits_{\mathrm{j}}^{\mathrm{T}} \left| \frac{\hat{\mathrm{Y}}_{\mathrm{i,j}} - \mathrm{Y}_{\mathrm{i,j}}}{\mathrm{Y}_{\mathrm{i,j}}} \right| \tag{27}$$

This article contrasts the performance of MVSC model with 5 mainstream models.

1) ARIMA: The ARIMA model is a classic time series analysis method used to predict future time series data. It combines three parts: AR model, difference operation, and moving average model. Model parameters (p, d, q) are set to (3,1,2) through data analysis.
2) GBRT: Gradient boosting regression tree model. It is an ensemble learning method that uses serial training of multiple decision tree models for prediction. At each step, the model will attempt to correct the errors in the previous step to gradually reduce the prediction error. The main parameter settings are as follows: The number of estimators is 100, the learning_rate is 1, the max-depth is 3, and the min_samples_leaf is 2.
3) LSTM [29]: This model is a variant of RNN, aimed at solving problems such as vanishing and exploding gradients in the original RNN model, especially suitable for tasks that require processing long sequence data. The parameter settings are as follows: The num_layers is 1, the hidden_size is 64, and the learning rate is 0.001.
4) STGCN: STGCN is a deep learning model for processing spatiotemporal data that combines the characteristics of (GCN) and spatiotemporal data, effectively capturing complex relationships and patterns in spatiotemporal data. The channels of three layers in ST-Conv block are 64, 16, 64 respectively, the graph and temporal convolution kernel size are set to 3, the input time step is 12, the batch size is 50, and the learning rate is 0.001.
5) Informer [12]: It is a deep learning model used for spatiotemporal sequence prediction, specifically designed to handle long sequence prediction problems. The model parameters are set as: The seq_len is 96, the label_len is 48, the enc_in, dec_in and c_out is 7, the n_head is 8 etc.

4.3. Contrast experiment

The performance comparison experiment with the benchmark model is carried out in this section. Tables 1 and 2 respectively list the experimental results on the data set of METR-LA and PEMS-BAY, from which the following conclusions can be drawn: On the two datasets, the proposed model performs better than the other five models in short-, medium-, and long-term prediction, except that the MAE of 30 min prediction on the METR-LA dataset is slightly worse than that of the Informer model. The ARIMA model has the worst overall performance on the two datasets, especially on the METR-LA dataset, where the performance deteriorates as the prediction step size increases. GBRT is an integrated model with good performance, and its performance is better than LSTM. The reason why LSTM is worse than STGCN, Informer and MVSC is the lack of mining the correlation features of sequence space. Because it is iteratively premeasured forward, the performance deteriorates with the increase of the predicted step size. STGCN is based on graph convolutional network to realize spatial correlation mining. Informer and MVSC directly embed input sequences as high-dimensional feature vectors, and realize spatiotemporal feature mining based on high-dimensional feature vectors, without strictly distinguishing the relationship between sequences. Judging by the results of the experiment, this strategy worked better. Compared with Informer and other models, MVSC relies on convolutional design as a time series prediction model. By setting different volume

Table 1
Comparison of traffic flow predictive performance on the METR-LA dataset

	15 min			30 min			60 min		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	3.35	6.33	9.13%	3.79	7.25	10.45%	4.37	8.72	13.18%
GBRT	2.89	5.84	7.22%	3.47	6.64	9.77%	4.07	7.89	11.26%
LSTM	3.34	7.15	8.13%	4.14	8.13	10.57%	5.03	9.79	12.19%
STGCN	2.94	5.65	7.64%	3.15	6.43	8.93%	4.58	8.42	10.65%
Informer	2.76	5.67	7.05%	3.08	6.32	8.82%	4.54	8.15	10.33%
MVSC	2.66	5.44	6.89%	3.13	6.18	8.44%	4.02	7.99	9.74%

Table 2
Comparison of predictive performance of the PEMS-BAY dataset

	15 min			30 min			60 min		
Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	2.05	4.18	4.82%	2.99	4.58	5.32%	2.39	5.02	6.72%
GBRT	1.38	2.99	3.03%	1.82	3.97	3.92%	2.18	4.89	5.93%
LSTM	1.49	3.13	3.21%	1.99	4.15	4.32%	2.72	5.52	7.03%
STGCN	1.34	2.97	2.99%	1.89	4.18	4.21%	2.59	5.78	6.94%
Informer	1.43	3.09	3.08%	1.72	4.02	3.99%	2.17	4.89	5.65%
MVSC	1.22	2.88	2.54%	1.33	3.01	2.98%	2.02	4.25	4.89%

nuclei and introducing channel attention, the final model is superior to Informer and other models.

4.4. Ablation experiment

Ablation experiments are performed in this section to verify the validity of the channel convolution and multi-view schemes used in the MVSC model. The following model variants were taken for comparison.

1) SSTC: single-view model, that is, only one convolution module is used in the spatiotemporal representation learning module. In the experiment, this paper only selected convolution kernel size 8 and convolution kernel size 12 for the experiment, which were denoted as SSTC-8 and SSTC-12 respectively.
2) MVSC/CA: Remove the Channel Attention module (CA for short). All variant model hyperparameters remain consistent with the original model. The results of the ablation experiment are shown in Table 3, and the following conclusions can be drawn: (1) The MVSC model is better than the other three variants; (2) From the results of MVSC/CA, it has the worst effect, which indicates that the channel attention module plays a significant role in MVSC. Due to the merging operation of multi-view space-time features, the channel attention module can effectively filter useful information and thus improve the model performance.

4.5. Hyperparameter analysis experiment

The MVSC model has two important hyperparameters: the multi-view combination mode and the hidden layer dimension. In Section 3.4, the ablation experiment shows that the multi-view approach is beneficial to spatiotemporal feature representation learning. This section focuses on the effects of different multi-view combination modes on model performance.

In Figure 5, the model's experimental results on the METR-LA and PEMS-BAY datasets are displayed, forecasting the traffic flow for the next 60 min. Only 4 combination modes were used in the

Table 3
Ablation result dataset

		15 min			30 min			60 min		
Dataset	Model	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	SSTC-8	2.72	5.63	7.02%	3.32	6.79	9.22%	4.63	8.33	11.24%
	SSTC-12	2.75	5.53	7.14%	3.29	6.82	9.35%	4.59	8.28	11.32%
	MVSC/CA	2.87	5.73	7.46%	3.42	6.98	9.43%	4.94	8.78	12.13%
	MVSC	2.66	5.44	6.89%	3.13	6.18	8.44%	4.02	7.99	9.74%
PEMS-BAY	SSTC-8	1.45	3.14	3.25%	1.97	4.13	4.29%	2.43	5.34	7.12%
	SSTC-12	1.52	3.21	3.29%	1.92	4.21	4.33%	2.39	5.32	7.04%
	MVSC/CA	1.57	3.19	3.38%	2.04	4.36	4.79%	2.62	5.53	7.22%
	MVSC	1.21	2.88	2.53%	1.33	3.02	2.99%	2.02	4.25	4.89%

Figure 5
Impact of multi-view approach (a) METR-LA dataset and (b) PEMS-BAY dataset

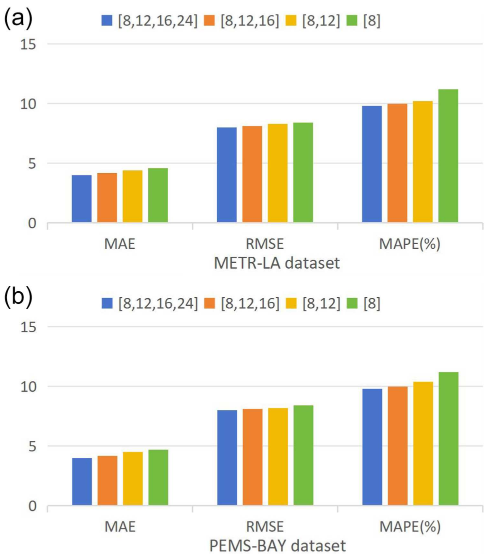
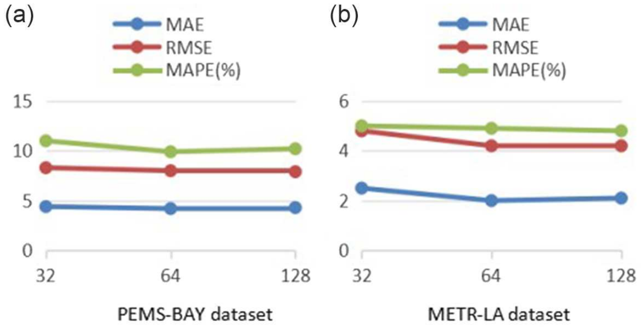


Figure 6
Influence of embedding dimensions on model performance (a) METR-LA dataset and (b) PEMS-BAY dataset



experiment. According to the experimental results, the model effect is gradually improved with the increase of view combination.

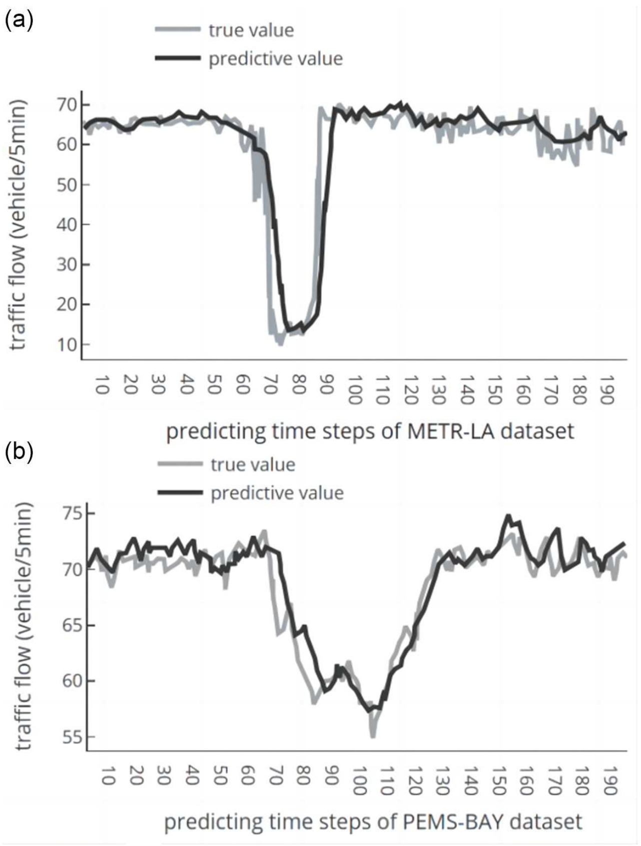
Figure 6 shows the performance table of the model under different hiding layer dimensions, predicting the traffic flow in the future 60 min. This paper solely examines the influence of the hidden layer dimension on the model without delving into the hidden layer dimensions of individual neural networks within the model.

Therefore, the hidden layer dimension of all networks in the experiment is set the same. As can be seen from the figure, the performance of the model is improved to some extent with the increase of the dimension of the hidden layer. However, the model performance is not significantly improved when the dimensions are 128 and 64. Therefore, in the experiment, the hidden layer dimension of each network layer of the MVSC model is set to 64.

4.6. Prediction examples

Figure 7 shows the predicted examples and true values of MVSC respectively.

Figure 7
Prediction results of dataset (a) Predicting time steps of METR-LA dataset and (b) Predicting time steps of PEMS-BAY dataset



It can be seen that MVSC is relatively accurate in predicting traffic flow on two datasets and has a good predictive effect on sudden changes in traffic flow.

5. Conclusion

This paper presents the development of the MVSC traffic flow prediction model based on convolutional networks. In the exploration of spatiotemporal features for traffic flow analysis, MVSC incorporates multiple spatiotemporal representation learning modules to discover various cycle patterns in traffic flow. Each module conducts local spatiotemporal feature extraction via one-dimensional convolution, followed by global spatiotemporal feature extraction with causal convolution. The prediction module integrates a channel attention mechanism to enhance the utilization of spatiotemporal features. The model has significantly improved the problem of insufficient utilization of spatiotemporal features in the previously mentioned models such as LSTM. The channel attention mechanism in the prediction layer has better enhanced the model's utilization rate of spatiotemporal features. The representation learning module has effectively learned the data features under different periodic patterns. The method of first local and then global spatiotemporal feature exploration has reduced the prediction error and interference.

Extensive experiments were conducted using real traffic flow datasets from METR-LA and PEMS-BAY, demonstrating the excellent performance of MVSC in traffic flow prediction tasks. And the forecasting method employed in the study is direct multi-step; on each dataset, the existing model shows particularly significant improvements in mid- to long-term predictions, with nearly a 1.2% enhancement in MAPE compared to traditional machine models or subsequent emerging models (such as STGCN). However, it also demonstrates corresponding shortcomings in short-term forecasts, with about a 0.4% gap compared to STGCN. The effectiveness of the multi-view mode and the channel attention mechanism was verified through ablation experiments. Additionally, hyperparameter analysis experiments were carried out, discussing the impact of multi-view combinations and hidden layer dimensions on the model. A limitation of this study is that sensor failures in transportation systems cause data loss. Therefore, this paper will further explore how to conduct spatiotemporal feature extraction on traffic flow datasets with missing values and design effective traffic flow prediction models based on convolutional networks.

Conflicts of Interest

The author declares that she has no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Xiaoyan Zhu: Conceptualization, Methodology, Software, Validation, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Lv, Z., Li, J., Dong, C., & Xu, Z. (2023). DeepSTF: A deep spatial-temporal forecast model of taxi flow. *The Computer Journal, 66*(3), 565–580. https://doi.org/10.1093/comjnl/bxab178

[2] Wang, Y., Zhao, A., Li, J., Lv, Z., Dong, C., & Li, H. (2023). Multi-attribute graph convolution network for regional traffic flow prediction. *Neural Processing Letters, 55*(4), 4183–4209. https://doi.org/10.1007/s11063-022-11036-9

[3] Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(9), 11121–11128. http://dx.doi.org/10.1609/aaai.v37i9.26317

[4] Wu, Y. Y., Zhao, L. N., Yuan, Z. X., & Zhang, C. (2023). CNN-GRU ship traffic flow prediction model based on attention mechanism. *Journal of Dalian Maritime University, 49*(1), 75–84.

[5] Huang, F., Yi, P., Wang, J., Li, M., & Peng, J. (2022). Time-series forecasting with shape attention. In *IEEE International Conference on Systems, Man, and Cybernetics,* 3299–3304. https://doi.org/10.1109/SMC53654.2022.9945543

[6] Yuan, H., Yu, H., Gui, S., & Ji, S. (2023). Explainability in graph neural networks: A taxonomic survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(5), 5782–5799. https://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3204236

[7] Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys, 55*(6), 109. https://doi.org/10.1145/3530811

[8] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning,* 27268–27286.

[9] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence, 35*(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[10] Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2021). An attentive survey of attention models. *ACM Transactions on Intelligent Systems and Technology, 12*(5), 1–32. https://doi.org/10.1145/3465055

[11] Yu, Q., Chen, Y., Zhang, Q., Li, L., & Ma, W. (2021). Short-term traffic flow prediction based on IWOA-WNN. In *33rd Chinese Control and Decision Conference,* 899–904. https://doi.org/10.1109/CCDC52312.2021.9601915

[12] Yu, B., Yin, H., & Zhu, Z. C. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence,* 3634–3640.

[13] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of the 35th International Conference on Neural Information Processing Systems,* 22419–22430.

[14] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G. J., & Xiong, H. (2020). Spatial-temporal transformer networks for traffic flow forecasting. *arXiv Preprint: 2001.02908*.

[15] Chen, B., Guo, W., Tang, R., Xin, X., Ding, Y., He, X., & Wang, D. (2020). TGCN: Tag graph convolutional network for tag-aware recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management,* 155–164. https://doi.org/10.1145/3340531.3411927

[16] Yang, F. M. (2020). Review of traffic flow prediction methods based on artificial neural networks. *Journal of Highway Transportation Science and Technology, 37*, 130–135.

[17] Chen, G., & Zhang, J. W. (2024). Intelligent transportation systems: Machine learning approaches for urban mobility in smart cities. *Sustainable Cities and Society, 107*, 105369. https://doi.org/10.1016/j.scs.2024.105369

[18] Wang, B., Long, Z., Sheng, J., & Zhong, Q. (2024). Spatial–temporal similarity fusion graph adversarial convolutional networks for traffic flow forecasting. *Journal of the Franklin Institute, 361*(17), 107299. https://doi.org/10.1016/j.jfranklin.2024.107299

[19] Miao, X., Wang, Z. Y., Wu, B., Yang, H., & Wang, Y. L. (2020). Bi-layer BPNN prediction model for bus arrival time considering preceding segment state. *Journal of Transportation Systems Engineering and Information Technology, 20*(2), 127–133.

[20] Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., & Feng, X. (2020). Multi-range attentive bicomponent graph convolutional network for traffic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(4), 3529–3536. https://doi.org/10.1609/aaai.v34i04.5758

[21] Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework

for spatial-temporal network data forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(1), 914–921. https://doi.org/10.1609/aaai.v34i01.5438

[22] Cheng, Z., Jian, S., Rashidi, T. H., Maghrebi, M., & Waller, S. T. (2020). Integrating household travel survey and social media data to improve the quality of od matrix: A comparative case study. *IEEE Transactions on Intelligent Transportation Systems, 21*(6), 2628–2636. https://doi.org/10.1109/TITS.2019.2958673

[23] Ghosh, P., Yao, Y., Davis, L., & Divakaran, A. (2020). Stacked spatio-temporal graph convolutional networks for action segmentation. In *IEEE Winter Conference on Applications of Computer Vision,* 565–574. https://doi.org/10.1109/WACV45572.2020.9093361

[24] Chao, Y., Wang, C., Wang, X., & Gao, Y. (2020). Urban road short-term traffic flow prediction based on spatio-temporal node selection and deep learning. *Journal of Computer Applications, 40*(5), 1488–1493. https://doi.org/10.11772/j.issn.1001-9081.2019091568

[25] Zhang, J., Zhao, S., Wang, Y., & Zhu, X. (2019). Improved social emotion optimization algorithm for short-term traffic flow forecasting based on back-propagation neural network. *Journal of Shanghai Jiaotong University (Science), 24*(2), 209–219. https://doi.org/10.1007/s12204-019-2055-9

[26] Guo, Z., Zhao, X., Chen, Y., Wu, W., & Yang, J. (2019). Short-term passenger flow forecast of urban rail transit based on GPR and KRR. *IET Intelligent Transport Systems, 13*(9), 1374–1382. https://doi.org/10.1049/iet-its.2018.5530

[27] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation, 31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

[28] Shewalkar, A., Nyavanandi, D., & Ludwig, S. A. (2019). Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research, 9*(4), 235–245. http://dx.doi.org/10.2478/jaiscr-2019-0006

[29] Kang, Z., Xu, H., Hu, J., & Pei, X. (2019). Learning dynamic graph embedding for traffic flow forecasting: A graph self-attentive method. In *IEEE Intelligent Transportation Systems Conference,* 2570–2576. https://doi.org/10.1109/ITSC.2019.8917213

How to Cite: Zhu, X. (2024). Multi-view Traffic Flow Prediction Model Based on Spatiotemporal Convolution. *Journal of Computational and Cognitive Engineering*, *3*(4), 395–403. https://doi.org/10.47852/bonviewJCCE42022914