Multi-Sensor Data Fusion for Short-Term Traffic Flow Prediction: A Novel Multi-Channel Data Structure Integrated with Mixed-Pointwise Convolution and Channel Attention Mechanism

Ruijun Feng[1(✉)] and Mingzhou Chen[2]

[1] Zhejiang University of Finance and Economics, Hangzhou, China
fengruijun@zufe.edu.cn
[2] Tongji University, Shanghai, China
2010361@tongji.edu.cn

Abstract. Accurate short-term traffic flow prediction is critical to improving the reliability and efficiency of intelligent transportation systems. However, the complex spatio-temporal characteristics of traffic flow pose a great challenge. The latest methods usually use a multi-sensor data fusion approach to learn the spatio-temporal information. First, it represents the traffic flow data collected from different sensors as a single-channel data structure. Then, the single-channel data structure combined with the multi-branch feature fusion strategy is used to learn the periodic dependencies (recent, daily, and weekly). However, these branches add a massive usage of parameters, which tends to overparameterize the prediction model, resulting in model overfitting and performance degeneration. To address these issues, a novel deep learning-based prediction model is proposed, which consists of three components. First, a new multi-channel data structure is proposed to efficiently reconstruct periodic dependencies of traffic data. Then, a new mixed-pointwise convolution method is proposed to extract spatio-temporal correlations and periodic dependencies of traffic data without over-parameterization and information loss. Last, an improved channel attention mechanism is employed to quantify the contributions of different channels. Extensive experiments are conducted on two real-world traffic datasets. The results demonstrate the proposed method consistently outperforms other baseline methods and has strong robustness in different settings.

Keywords: Traffic flow prediction · Multi-channel data structure · 3D convolution · Pointwise convolution · Channel attention

1 Introduction

Short-term traffic flow prediction aims to predict the traffic flow for the next five to 30 min based on the historical traffic data. And it's of great significance to build the Intelligent Transportation Systems (ITS). Traditional prediction methods such as Historical Average (HA) [17], Auto-Regressive Integrated Moving Average (ARIMA) [17],

E. Pimenidis et al. (Eds.): ICANN 2022, LNCS 13532, pp. 731–742, 2022.
https://doi.org/10.1007/978-3-031-15937-4_61

and Support Vector Regression (SVR) [18], only consider the intra-dependencies (i.e., temporal correlation across a single sequence), but ignoring the inter-information (i.e., spatio-temporal correlations across multiple sequences).

Recently, a modern solution that adopts the multi-sensor data fusion method has arisen. First, it converts traffic data collected from multiple sensors into different data formats that can represent spatial dependency. Then, deep learning-based methods are used to capture the spatio-temporal correlations, which include two types of methods: Convolutional Neural Network (CNN)-based methods and Recurrent Neural Network (RNN)-based methods. RNN-based methods [6] are good at capturing temporal dependencies but fail to consider spatial dependency without additional help, and usually suffered from low parallel efficiency and vanishing gradient due to recursive design. CNN-based methods, on the other hand, can capture the spatio-temporal correlations [2, 4, 15, 19] with high parallel efficiency and no vanishing gradient. In our previous work, CNN is used for its superior ability in capturing spatio-temporal correlations [20].

Traditional CNN-based methods reconstruct the traffic data into a two-dimensional matrix and apply a 2D-CNN for feature extraction [15]. This matrix stacks multiple one-dimensional sequences vertically, making spatial information very ambiguous because it can't represent the real geographic distribution of different sections. Further researches improve it by using a three-dimensional tensor, each two-dimensional matrix is a snapshot of the transportation network. And they further enhance temporal correlation by using three parallel tensors with three parallel branches, corresponding with three types of periodic dependency: recent, daily, and weekly (hereinafter referred it as multi-branch feature fusion strategy). In the beginning, a Spatio-Temporal Residual Network (ST-ResNet) [19] based on 2D convolution is proposed. But due to the limitations of 2D convolution, the temporal information will lose after the first layer. To improve that, a multiple local 3D CNN Spatio-Temporal Residual Network (LMST3D-ResNet) [4] is proposed by replacing the 2D convolution with 3D convolution. Compared with 2D convolution, 3D convolution can preserve more temporal information.

However, the aforementioned methods still have a huge research gap. First of all, the multi-branch feature fusion strategy is highly inefficient. It's not worth tripling the parameters or even more just to account for periodic dependencies, which is likely to cause over-parameterization. Second, they all suffered from information loss to some extent, due to the usage of max pooling layers or 2D convolution. Third, they all ignored the channel inter-dependencies, which are useful in terms of CNN-based methods.

To tackle these challenges, a novel multi-channel data structure integrated with mixed-pointwise convolution and channel attention mechanism (CAMPConv-MC) is proposed. The main contributions of this article are summarized as follows:

1. A new multi-channel data structure is proposed by reconstructing traffic data into a four-dimensional tensor. This data structure can make full use of the channel parallelism in CNN while representing the periodic dependencies of traffic data.
2. A new mixed-pointwise convolution method is proposed that integrates 3D convolution, 2D convolution, and their variants pointwise convolution to extract spatio-temporal correlations. No max pooling layer or 2D convolution is used in the middle, thus alleviating the information loss. And it removes the multi-branch feature fusion

strategy and fully connected (FC) layers, hence over-parameterization is avoided as well.
3. A channel attention mechanism is adapted from the squeeze and excitation (SE) unit [8] and employed to learn the channel inter-dependencies of 3D convolution with a mild parameter usage.

2 Related Work

2.1 Traditional Traffic Prediction Methods

Traditional traffic prediction methods include parametric methods and non-parametric methods. Some typical examples of parametric methods are HA and ARIMA [17]. These methods achieve satisfactory performance on short series that are stational and univariate. But the strong assumptions about data are not suitable for non-linear traffic data. These limitations have been improved by non-parametric methods. A typical example is SVR [18]. SVR uses a kernel function to project the traffic data to high dimensional space. This reconstruction makes non-linear traffic data linearly separable with hyperplanes. Other methods such as Bayesian Network [12] and K-Nearest Neighbor [12] all achieved a better forecast error by considering spatial dependency. However, these non-parametric methods still require some human intervention like feature engineering. In comparison with them, deep learning-based methods learn the features on their own with no human help required.

2.2 Deep Learning-Based Traffic Prediction Methods

In the age of big data, deep learning-based methods like Deep Belief Networks (DBF) [9], Stacked Auto Encoder (SAE) [14], Gated Recurrent Unit (GRU) [6], and Long Short-Term Memory (LSTM) [6] have been proposed for traffic data prediction. DBF and SAE focus on spatial dependency but ignore the long-term patterns while GRU and LSTM are the opposite. In addition, RNN-based methods (i.e., LSTM and GRU) have poor parallel efficiency and vanishing gradient problems as they are trained [2]. Subsequent methods have overcome these shortcomings with CNN-based models and multi-sensor data fusion to reconstruct and capture the spatio-temporal correlations. At an early stage, Ma et al. [15] proposed a traditional 2D-CNN with two-dimensional matrix. Later, Zhang et al. [19] extended it into three-dimensional tensor and proposed the ST-ResNet. However, 2D convolution used in ST-ResNet can't preserve temporal information after the first layer. To improve it, Chen et al. [4] proposed an LMST3D-ResNet by replacing the 2D convolution with 3D convolution and introducing the 3D max pooling layers and FC layers. In addition, ST-ResNet and LMST3D-ResNet adopt a multi-branch feature fusion strategy to enhance the modeling of temporal correlation. But this strategy and FC layers use a huge number of parameters, which is likely to cause over-parameterization that triggers overfitting and performance degeneration. What's more, these CNN-based methods with max pooling layers condense too much information when extracting features, resulting in information loss. Moreover, they all neglect the importance of channel inter-dependencies, which is crucial in CNN-based methods.

2.3 Pointwise Convolution and Channel Attention Mechanism

Pointwise convolution was proposed in fully convolutional methods like the Xception network [5]. Pointwise convolution refers to convolution with a kernel size of one. It's used to reduce channels before expensive filters while learning the channel interdependencies. Meanwhile, a channel attention mechanism was proposed to refine informative filter output. Chen et al. [3] used global mean pooling with softmax function and FC layers to find the informative channels. Liu et al. [13] applied it in traffic data prediction using multiple two-dimensional matrixes. However, apart from the drawbacks of the two-dimensional matrix, the FC layers are very computational expensive as the channels increase. SE unit [8] improved it with more flexible squeeze and excitation operations. Inspired by them, a mixed-pointwise convolution integrated with a channel attention mechanism is proposed to reduce computations and boost performance.

3 Methodology

3.1 Multi-channel Data Structure

In this article, a transportation network is defined as regular raster data [1] to fuse the spatio-temporal information collected from multiple traffic sensors. Apart from spatio-temporal correlations, the proposed method adds periodic dependencies into account.

Definition 1: Rasterization. The transportation network is partitioned into $I \times J$ grids based on latitude and longitude. Each sensor records at time point t are assigned to the closest grid (i, j) and averaged into a scalar denoted as $x_t^{i,j}$.

Definition 2: Spatio-Temporal Raster Data. First, a two-dimensional matrix denoted as $S_t \in \mathbb{R}^{I \times J}$ is used to represent the spatial information at time point t, as defined in Eq. (1):

$$S_t = \{x_t^{i,j} | i \in I, j \in J\} \tag{1}$$

Then, the temporal information is represented by $T_p \in \mathbb{R}^{d \times I \times J}$ as defined in Eq. (2) where d refers to subsequence length and p is sampling period.

$$T_p = \{S_{t-d \times p}, S_{t-(d-1) \times p}, ..., S_t\} \tag{2}$$

Definition 3: Multi-Channel Data Structure. The multi-channel data structure is denoted as $X_C \in \mathbb{R}^{C \times d \times I \times J}$, composed of C types of periodic dependency: recent, daily, and weekly. X_C can be rewritten as $X_C = \{T_r, T_d, T_w\}$. For each tensor T, d is set to the same, and p is set to five minutes, 24 h, and one week respectively.

Definition 4: Traffic Flow Prediction. As defined in Eq. (3). Given X_C, the goal is to predict the value at the time point $t + \Delta t$ denoted as $\widehat{X}_{t+\Delta t}$, where Δt is the forecast time interval, and θ is the trainable parameters of the proposed CAMPConv.

$$\widehat{X}_{t+\Delta t} = f_\theta(X_C) \tag{3}$$

3.2 Mixed-Pointwise Convolution Integrated with Channel Attention Mechanism

In this section, a novel CAMPConv is proposed which contains two components: convolutional unit, and SE unit as shown in Fig. 1(b) and Fig. 1(c). Figure 1(a) is the overall structure of the proposed CAMPConv-MC. First, the input unit uses a 3D pointwise convolutional unit with 64 filters is used to project the proposed multi-channel data structure X_C to a high-dimensional channel space. Then, the backbone uses multiple 3D convolutional units (same padding) with 64 filters and SE units are employed to extract the spatio-temporal correlations. After this, the output unit uses a 3D pointwise convolutional unit with one filter implemented to compress the channel space into one. At last, the 3D channel axis is removed so that a 2D pointwise convolutional unit can be employed to extract temporal information of the same region and make predictions.

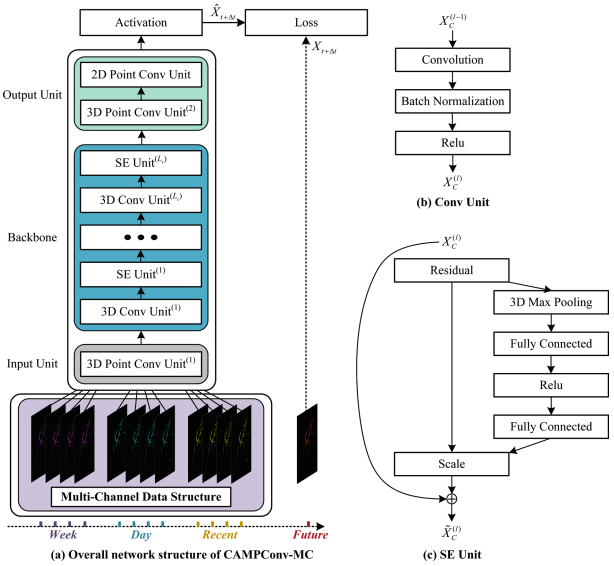

Fig. 1. Network structure and the details of the proposed CAMPConv-MC. (3D Conv Unit: 3D Convolutional Unit; 3D Point Conv Unit: 3D Pointwise Convolutional Unit; 2D Point Conv Unit: 2D Pointwise Convolutional Unit; SE Unit: Squeeze and Excitation Unit)

Mixed-Pointwise Convolution. Mixed-pointwise convolution is a hybrid method that adopts 3D convolution, 2D convolution, and their variants pointwise convolution. They are denoted as convolutional units as demonstrated in Fig. 1(b). First, a convolutional layer is used to extract spatio-temporal correlations. Then, a batch normalization [10] layer is implemented to avoid internal covariate shift. Finally, a relu function is used to add nonlinearity. As defined in Eq. (4), $X_C^{(l-1)} \in \mathbb{R}^{C_{l-1} \times d \times I \times J}$ denotes the input of the l^{th} 3D convolutional layer, which is the output of the upper layer with C_{l-1} channels. $W^{(l)}$ and $b^{(l)}$ are the weights and bias of the l^{th} 3D convolutional layer, and * denotes the operation of 3D convolution. BN denotes the batch normalization operation and δ

denotes the relu function. This design preserves the temporal information by keeping the sequence length d unchanged during 3D convolution.

$$X_C^{(l)}=\delta(BN(W^{(l)} * X_C^{(l-1)} + b^{(l)})) \tag{4}$$

After 3D convolution, 2D pointwise convolution is used to extract temporal information for prediction. As defined in Eq. (5), L refers to the last 3D pointwise convolutional units as shown in Fig. 1(a). W and b refer to the weights and bias of the 2D convolutional layer, $\circledast$ denotes the 2D convolutional operation.

$$\hat{X}_{t+\triangle t}=\delta(BN(W \circledast X_C^{(L)} + b)) \tag{5}$$

Squeeze and Excitation Unit. This article extends the traditional SE unit to meet the requirements of traffic prediction, as shown in Fig. 1(c). Unlike image data, spatio-temporal traffic raster data often contains many empty regions due to the sparsity of the transportation network. The global mean pooling used in traditional SE units will introduce too much noise. In this article, a simple adaption to 3D max pooling is used to pick out the most informative region in the spatio-temporal domain.

In the notion that follows, F_{sq} denotes squeeze operation and F_{ex} denotes excitation operation. $X_C^{(l)} \in \mathbb{R}^{C_l \times d \times I \times J}$ is the output of the l^{th} 3D convolutional unit, which can be rewritten as $X_C^{(l)} = \{x_1^{(l)}, x_2^{(l)}, ..., x_{C_l}^{(l)}\}$ where $x_c^{(l)}$ refers to the output of the c^{th} filter in l^{th} layer. The global spatio-temporal information is squeezed out using 3D max pooling, denotes as MAX. The element z_c of statistic $z \in \mathbb{R}^{C_l}$ is calculated by Eq. (6):

$$z_c = F_{sq}(x_c^{(l)}) = MAX\,(x_c^{(l)})\ \ c = 1, 2, ...C_l \tag{6}$$

After squeezing out the global spatio-temporal information, the excitation operation uses two FC layers to learn the non-linear interaction. To control network complexity, the first FC layer encodes z into a smaller space and the next FC layer decodes it back to the original space. The reduction ratio is denoted as r, and the weights of these FC layers are denoted as $W_1 \in \mathbb{R}^{\frac{C_l}{r} \times C_l}$, $W_2 \in \mathbb{R}^{C_l \times \frac{C_l}{r}}$. σ denotes the sigmoid function and δ denotes the relu function. After excitation operation, a vector $s \in \mathbb{R}^{C_l}$ is calculated as:

$$s = F_{ex}(z, W) = \sigma(W_2\delta(W_1 z)) \tag{7}$$

In the end, as defined in Eq. (8), channel-wise multiplication and residual link [7] are conducted between s and $X_C^{(l)}$ to get the refined feature maps denoted as $\tilde{X}_C^{(l)} \in \mathbb{R}^{C_l \times d \times I \times J}$. It can be rewritten as $\tilde{X}_C^{(l)} = \{\tilde{x}_1^{(l)}, \tilde{x}_2^{(l)}, ..., \tilde{x}_{C_l}^{(l)}\}$, where $\circ$ is the element-wise multiplication and $\tilde{x}_c^{(l)}$ refers to the refined output of the c^{th} filter in the l^{th} layer.

$$\tilde{x}_c^{(l)} = s_c \circ x_c^{(l)} + x_c^{(l)}\ \ c = 1, 2, ...C_l \tag{8}$$

4 Experiments

4.1 Computing Environment and Datasets

In terms of the computing environment, the following experiments were conducted on a server with 16 physical cores (Intel Xeon Silver 4110 @ 2.10 GHz) and two

GPUs (RTX 2080Ti). The software environment uses python 3.8.5 with pytorch 1.7.1, pytorch-lightning 1.1.8, and statsmodels 0.12.2 with Windows10 20H1 to build models.

This article uses two high way traffic datasets collected by the Caltrans Performance Measurement System (PeMS). PeMS collects real-time data samples every 30-s and aggregated them into 5-min time interval. Table 1 shows the details of these datasets. The raster sizes are set to (42, 34) and (20, 36) so that each grid is 5 km $\times$ 5 km. At first, detectors with any empty records are eliminated from the dataset. After converting the traffic data into spatio-temporal traffic raster data, min-max normalization is adopted to scale the value between zero to one. Then, the raster data is converted to the multi-channel data structure. Finally, the dataset is divided into training data and testing data under the ratio of 8:2, then 20% of the training data is cut out as validation data.

Table 1. Descriptions of the two high way traffic datasets

Dataset	PeMSD4	PeMSD7
Location	San Francisco Bay Area	District 7 of California
Number of detectors	3796	4817
Time span	1st Jun 2017–30th Jun 2017	1st Jun 2017–30th Jun 2017
Time interval	5-min	5-min
Raster size	(42, 34)	(20, 36)
Available time points	8640	8640

4.2 Baselines and Benchmarks

The proposed method is compared with the following six baseline methods:

- HA: The predicted values are estimated with four historical time points.
- ARIMA: A classic statical model for time series prediction. It uses the order of (0, 1, 1) according to the previous study [17].
- CNN: CNN uses two 2D convolutional layers (valid padding) with a kernel size of three, a filer num of 128 and 64, and a stride of one; two max pooling layers are used with a kernel size of two and a stride of two; and one FC layer with serval neurons.
- ConvLSTM: Convolutional LSTM is a variation of LSTM which is good at capturing spatio-temporal correlations [16]. It uses three layers (same padding) with a filter num of six, a kernel size of three, and a stride of two.
- ST-Resnet: A fully convolutional method uses 12 residual units; each contains a 2D convolutional layer (same padding) with a filter num of 64, a kernel size of three, and a stride of one. And a multi-branch feature fusion strategy is also used.
- LMST3D-ResNet: An improved version of ST-Resnet. It uses three residual units, each contains a 3D convolutional layer (same padding) with a filter num of 64, a kernel size of three, and a stride of one; and a 3D max pooling layer uses a kernel size of (2, 3, 3), the stride of one, and a padding size of (0, 1, 1). Two FC layers are used in the last. A dropout rate of 20% is used to avoid overfitting.

For fairness, all methods use the same computing environment with four historical time points to make future predictions. All deep learning-based methods share the same hyperparameter setting with a learning rate of 1e-4 and a batch size of 64. They are trained for 200 epochs by Adam optimizer [11] with $l1$ loss as defined in Eq. (9), where I and J are the width and height of the grid map, $\hat{x}^{i,j}_{t+\Delta t}$ refers to predicted value and $x^{i,j}_{t+\Delta t}$ is ground truth. The model with a minimal loss value is kept as final model.

$$Loss = \frac{1}{I \times J} \sum_{j=1}^{J} \sum_{i=1}^{I} \left| x^{i,j}_{t+\Delta t} - \hat{x}^{i,j}_{t+\Delta t} \right| \tag{9}$$

Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are used to measure the forecast performance. As defined in Eq. (10) and Eq. (11), n is the number of ground truths at time point $t + \Delta t$, $\hat{x}^{i}_{t+\Delta t}$ and $x^{i}_{t+\Delta t}$ are the rescaled predicted value and the corresponding ground truth.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x^{i}_{t+\Delta t} - \hat{x}^{i}_{t+\Delta t})^2} \tag{10}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| x^{i}_{t+\Delta t} - \hat{x}^{i}_{t+\Delta t} \right| \tag{11}$$

5 Results and Analysis

This section shows the results in predicting the traffic flow in the next 15-min with the grid size of 5 km × 5 km. The proposed CAMPConv-MC uses three 3D convolutional units and three SE units. All convolutional layers (same padding) use 64 filters, a kernel size of three, and a stride of one. The reduction ratio r for SE units is four.

5.1 Experimental Results on PeMSD4

Comparison with Baseline Methods. The results on the PeMSD4 dataset are listed in Table 2, where the proposed CAMPConv-MC beats all other methods. The worst are HA and ARIMA due to the neglect of spatial dependencies. Despite CNN and ConvLSTM considering spatial dependencies, they have a significant drawback as they only consider spatial dependencies of nearby regions. On the contrary, ST-Resnet considers the long-distance spatial dependency using residual units. But 2D convolution fails to preserve temporal information for it treats the temporal axis as multiple channels. LMST3D-ResNet replaces 2D convolution with 3D convolution to preserve more temporal information. However, LMST3D-ResNet suffered from over-parameterization and information loss due to the multi-branch feature fusion strategy and massive usage of max pooling layers, even regularization techniques like dropout can't fundamentally

Table 2. Performance comparison on PeMSD4 (The best values are marked in bold)

Methods	MAE	RMSE
HA	21.425	28.406
ARIMA	20.221	27.410
CNN	15.540	21.027
ConvLSTM	18.689	25.047
ST-Resnet	14.822	20.399
LMST3D-Resnet	13.948	19.798
CAMPConv-MC	**13.464**	**18.832**

solve it. The proposed CAMPConv-MC capture the correlations between different periodic dependencies and the channel inter-dependencies. It achieves the best performance without any max pooling layers or a massive usage of parameters. To investigate the effectiveness of each part, an ablation study is conducted in next section.

Ablation Study of CAMPConv-MC. This section tests a few variants of the CAMPConv-MC with a new metric called Time, which refers to the training time. A fully convolutional network made up of multiple 3D convolutional units (M3D) trained on a single-channel data structure is used as a baseline. M3D-CA denotes baseline plus SE unit, M3D-MP denotes baseline plus mixed-pointwise convolution, and M3D-MC denotes baseline plus multi-channel data structure. Expect for M3D-MP, all models use the kernel of size (2, 3, 3) and the padding of size (0, 1, 1) to compress temporal channels into one. Table 3 proves that every part of the proposed model is effective compared with M3D. When compared to M3D, M3D-MC has superior MAE and RMSE with nearly equal training times while not using any branch, demonstrating the high efficiency of the proposed data structure. M3D-MC has a larger RMSE but a substantially lower MAE when compared to M3D-MP. This showed that the spatio-temporal correlations can improve forecast performance in extreme conditions thus increasing the robustness. Additionally, M3D-CA outperforms M3D proves the necessity of considering the channel inter-dependencies and the validity of the improved SE units.

Table 3. Effects of different components of the proposed method

Methods	Time (Minutes)	MAE	RMSE
M3D	**9.666**	16.926	26.174
M3D-MC	9.672	14.789	24.034
M3D-CA	12.219	15.801	24.987
M3D-MP	22.858	15.173	20.876
CAMPConv-MC	27.146	**13.464**	**18.832**

5.2 Experimental Results on PeMSD7

Comparison with Baseline Methods. The results of PeMSD7 are similar to PeMSD4. As shown in Table 4, CAMPConv-MC consistently outperforms other baseline methods. Because the raster size of PeMSD7 is smaller than PeMSD4, the forecast errors on PeMSD7 are smaller than PeMSD4. The proposed CAMPConv-MC performs well on both datasets, yielding strong robustness under different raster sizes. To further prove the advantages of CAMPConv-MC, a scalability analysis is conducted in the next section.

Table 4. Performance comparison on PeMSD7

Methods	MAE	RMSE
HA	20.883	25.325
ARIMA	18.518	23.084
CNN	14.591	18.745
ConvLSTM	16.701	21.320
ST-Resnet	13.576	18.041
LMST3D-Resnet	13.497	17.752
CAMPConv-MC	**12.408**	**16.126**

Scalability Analysis. Table 5 and Table 6 show the results of RMSE under different grid sizes and time intervals. For fairness, one setting is changed while the other is kept unchanged. As shown in Table 5, forecast errors decrease as the grid size enlarges. This is because, given a fixed transportation network, a smaller grid size can generate more grids to be predicted. At the scale of 10 km $\times$ 10 km, ST-Resnet and LMST3D-Resnet don't vary a lot compared with CNN. This is due to the grid map becoming too small. At this scale, even CNN can discern the long-distance spatial dependencies. As shown in Table 6, forecast errors increase as time intervals enlarge. This is due to the fact that a larger time interval will result in more temporal uncertainty. For instance, when the time interval is 30-min, the increased temporal uncertainty narrowed the performance gap between LMST3D-Resnet and ST-Resnet. However, the proposed methods consistently outperform all other deep learning-based methods, displaying the highest robustness regardless of grid size and time interval.

Table 5. RMSE under different grid sizes (time interval of 15-min)

Methods	2.5 km × 2.5 km	5 km × 5 km	10 km × 10 km
CNN	22.266	18.745	15.810
ConvLSTM	25.330	21.320	19.065
ST-Resnet	25.332	18.041	15.384
LMST3D-Resnet	19.931	17.752	15.556
CAMPConv-MC	**18.524**	**16.126**	**15.242**

Table 6. RMSE under different time intervals (grid size of 5 km × 5 km)

Methods	5-min	15-min	30-min
CNN	18.244	18.745	19.981
ConvLSTM	19.822	21.320	23.840
ST-Resnet	18.093	18.041	18.645
LMST3D-Resnet	17.818	17.752	18.440
CAMPConv-MC	**17.030**	**16.126**	**17.069**

6 Conclusions and Future Work

In this article, a novel CAMPConv-MC is proposed for short-term traffic flow prediction. A novel multi-channel data structure is proposed to efficiently build periodic dependencies of traffic data. Then, a new mixed-pointwise convolution is proposed to capture the spatio-temporal correlations and periodic dependencies, and an improved channel attention mechanism adapted from traditional SE unit is used to learn the channel interdependencies. Extensive experiments on two real-world datasets show that the proposed method exceeds the state-of-the-art methods in robustness and performance.

However, there are some limitations to this article. As grid size goes up, the sparsification of grid maps is a problem. Moreover, the effects of external features and results on datasets from different countries are remains to be discussed in future work.

References

1. Atluri, G., Karpatne, A., Kumar, V.: Spatio-temporal data mining: a survey of problems and methods. ACM Comput. Surv. **51**(4), Article. No. 83, 1–41 (2018)
2. Bai, S.J., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic Convolutional and recurrent networks for sequence modeling. arXiv:1803.01271v2, https://arxiv.org/abs/1803.01271 (2018)
3. Chen, L., Zhang, H.W., Xiao, J., Nie, L.Q., Shao, J., Liu, W. et al.: SCA-CNN: spatial and channel-wise attention in convolutional networks for image captioning. In: Proceedings of the

2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21–26 July, pp. 6298–6306. Honolulu, USA (2017)
4. Chen, Y.B., Zou, X.F., Li, K.L., Li, K.Q., Yang, X.L., Chen, C.: Multiple local 3D CNNs for region-based prediction in smart cities. Inf. Sci. **542**, 476–491 (2021)
5. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21–26 July, pp. 1800–1807. Honolulu, USA (2017)
6. Fu, R., Zhang, Z., Li, L.: Using LSTM and GRU neural network methods for traffic flow prediction. In: Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), 11–13 November, pp. 324–328. Wuhan, China (2016)
7. He, K.M., Zhang, X.Y., Ren, S.Q., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27–30 June. pp. 770–778. Las Vegas, USA (2016)
8. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.H.: Squeeze-and-excitation networks. IEEE Trans. Pattern Anal. Mach. Intell. **42**(8), 2011–2023 (2020)
9. Huang, W.H., Song, G.J., Hong, H.K., Xie, K.Q.: Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. **15**(5), 2191–2201 (2014)
10. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning (ICML), 6–11 July, pp. 448–456. Lille, France (2015)
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR), 7–9 May, pp. 1–15. San Diego, USA (2014)
12. Kuang, L., Yan, H., Zhu, Y.J., Tu, S.M., Fan, X.L.: Predicting duration of traffic accidents based on cost-sensitive Bayesian network and weighted K-nearest neighbor. J. Intell. Transp. Syst. **23**(2), 161–174 (2019)
13. Liu, Q.C., Wang, B.C., Zhu, Y.Q.: Short-term traffic speed forecasting based on attention convolutional neural network for arterials. Comput.-Aided Civil Infrast. Eng. **33**(11), 996–1016 (2018)
14. Lv, Y.S., Duan, Y.J., Kang, W.W., Li, Z.X., Wang, F.Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
15. Ma, X.L., Dai, Z., He, Z.B., Ma, J.H., Wang, Y., Wang, Y.P.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)
16. Shi, X.J., Chen, Z.R., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In: Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS), 7–12 December, pp. 802–810, Montreal, Canada (2015)
17. Smith, B.L., Demetsky, M.J.: Traffic flow forecasting: comparison of modeling approaches. J. Transp. Eng. **123**(4), 261–266 (1997)
18. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. **5**(4), 276–281 (2004)
19. Zhang, J.B., Zheng, Y., Qi, D.K., Li, R.Y., Yi, X.W., Li, T.R.: Predicting citywide crowd flows using deep spatio-temporal residual networks. Artif. Intell. **259**, 147–166 (2018)
20. Zhang, S., Chen, Y., Zhang, W.Y., Feng, R.J.: A novel ensemble deep learning model with dynamic error correction and multi-objective ensemble pruning for time series forecasting. Inf. Sci. **544**, 427–445 (2021)