Neurocomputing 427 (2021) 169–178

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A temporal-aware LSTM enhanced by loss-switch mechanism for traffic flow forecasting

Huakang Lu [a], Zuhao Ge [a], Youyi Song [b], Dazhi Jiang [a,c], Teng Zhou [a,b,c,*], Jing Qin [b]

[a] Department of Computer Science, Shantou University, Shantou, China
[b] Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong
[c] Key Laboratory of Intelligent Manufacturing Technology (Shantou University), Ministry of Education, Shantou, China

ARTICLE INFO

Article history:
Received 26 April 2020
Revised 15 September 2020
Accepted 15 November 2020
Available online 28 November 2020
Communicated by Steven Hoi

Keywords:
Intelligent transportation system
Traffic flow modeling
Time series analysis
Deep learning
Noise-immune learning

ABSTRACT

Short-term traffic flow forecasting at isolated points is a fundamental yet challenging task in many intelligent transportation systems. We present a novel long short-term memory (LSTM) network enhanced by temporal-aware convolutional context (TCC) blocks and a new loss-switch mechanism (LSM) to carry out this task. Compared with conventional recurrent neural networks (RNN) or LSTM networks, the proposed network can capture much more distinguishable temporal features and effectively counteracting noise and outliers for more accurate prediction. The proposed TCC blocks, leveraging dilated convolution, produce an enlarged receptive field in temporal contexts, and formulate a temporal-aware attention mechanism to learn the complicated and subtle temporal features from the traffic flows. We further cascade multiple TCC blocks in the network to learn more temporal features at different scales. To deal with the noise and outliers, we propose a novel loss-switch mechanism (LSM) by combining the traditional mean square error loss and the generalized correntropy induced metric (GCIM), which is capable of effectively counteracting non-Gaussian disturbances. The whole network is trained in an end-to-end manner guided by the loss-switch mechanism. Extensive experiments are conducted on two typical benchmark datasets and the experimental results corroborate the superiority of the proposed model over state-of-the-art methods.

1. Introduction

Short-term traffic flow forecasting (STTFF) is a classical yet still challenging problem. It is a key component of various intelligent transportation systems, including navigation systems [1], traffic signal control systems [2], and traveler information systems [3].

Early studies employed model-driven approaches to solve this problem. Comparing with model-driven approaches, such as Kalman family filters [4–6] or time series models [7,8], data-driven approaches, especially latest deep neural networks, such as deep belief network (DBN) [9], stacked autoencoder (SAE) [10–12], recurrent neural network (RNN) [13,14], generative adversarial nets [15], and extreme learning machine [16,17] have greatly improved the accuracy of the forecasting performance. A more detailed literature review can be find in [18].

Recent works have further boosted the forecasting accuracy by the benefit of spatial information [19–22,21,23–25]. For example, Lv et al. [19] proposed a deep learning model that integrates both CNN and RNN to take the advantage of the spatial and temporal information of the road network. Yu et al. [20] developed a spatio-temporal graph convolutional network for road network topology-award forecasting task. Pan et al. [22,23] proposed a deep meta learning based model, termed ST-MetaNet+, to deal with the complex spatial diversity and temporal diversity of the spatio-temporal correlations of urban traffic. However, the context of imbalanced spatio-temporal sequences may lead to poor performances of the current deep neural network [24,26]. Wu et al. [24] proposed a hierarchical structured spatial-temporal transformer network to alleviate the data imbalance issue and preserve the semantic signal in a fully dynamic manner. Recently, some deep learning models proposed to capture the spatio-temporal correlations between urban regions, but trapped in the inefficiency in learning global spatial dependencies, and the overlooking latent region functions. Liang et al. [25] present a novel framework endowed by a local feature extraction module, a global context module, and a region-specific predictor to tackle these issues. Pan et al. [27] designed a novel framework by matrix factorization for spatio-temporal feature learning and region-specific prediction.

[*] Corresponding author at: Department of Computer Science, Shantou University, Shantou, China.

E-mail address: zhouteng@stu.edu.cn (T. Zhou).

https://doi.org/10.1016/j.neucom.2020.11.026
0925-2312/© 2020 Elsevier B.V. All rights reserved.

Huang et al. [21] developed a multi-view and multi-modal spatial temporal learning framework to address the issue of intra and inter region correlations, and latent cross-categorical dependencies. However, it is difficult, if not impossible, to apply spatial information to isolated points in a transportation network, such as the boundary points of a road network or the sparsely located detectors in transportation systems in developing countries [28,29]. At these points, we can only use temporal traffic information, making the forecasting task more challenging. It is estimated small variations (less than 1 vehicle/min) of traffic flow at the boundary points will cause the change of the speed for more than 10% in a road network [30]. In this regard, the forecasting accuracy at these points plays a key role for the success of the whole traffic flow forecasting systems. On the other hand, the effective and efficient learning of temporal dependencies can be integrated in other spatio-temporal forecasting applications.

STTFF at isolated points, in principle, requires a comprehensive understanding of both trend and seasonality of the traffic flow sequence, as well as an effective identification of impulsive noises. The regular traffic flow sequence is easily affected by exterior noises, such as unexpected accidents or manual traffic control. These irregular samples may disturb the normal training process of the deep neural networks. Thus, a more robust network to effectively prevent the disturbance of irregular samples is required. Extensive attempts have been dedicated to develop various deep networks for this task, particularly various RNNs [13,14,31]. Unfortunately, traditional RNNs suffer a major drawback of gradient vanishing or exploding, which led to the improved version, long short-term memory neural network (LSTM) [32]. The existing LSTM networks, however, still have some shortcomings for STTFF at isolated points. First, conventional LSTMs must wait the predecessor to complete the prediction of the former time interval, which may easily consume up the memory to store the intermediate results for the gates. Second, mean square error losses have usually been employed in conventional LSTMs, such as [19], which uses the net effect of amplifying the contribution of prediction errors that are far away from the mean value of the error distribution. In this regard, Gaussian distribute residuals is the key to the success of the conventional LSTMs for traffic flow forecasting. However, unexpected incidents, sudden changes of weather, or manual traffic flow control bring impulsive disturbances to the traffic flow [33], and the failure of the loop detectors may also produce large outliers [34,35]. To the end, the Gaussian assumption of the prediction residuals cannot be always insured, and hence the conventional LSTMs may deteriorate severely in these cases.

To address these shortcomings, we propose a novel LSTM enhanced by temporal-aware convolutional context (TCC) blocks and a new loss-switch mechanism (LSM) for traffic flow forecasting at isolated points. Our TCC blocks, leveraging dilated convolution [36], produce an enlarged receptive field in temporal contexts, and formulate a temporal-aware attention mechanism to learn complicated and subtle temporal features from the traffic flows. We further cascade multiple TCC blocks in the network to learn more temporal-aware features at different scales. We then feed the temporal features learned from the TCC blocks into the LSTM network to produce the forecasting results. In order to deal with the noise and outliers, we propose a novel loss-switch mechanism (LSM) by combining the traditional mean square error loss and the generalized correntropy induced metric (GCIM), which is capable of effectively counteracting non-Gaussian disturbances. The whole network is trained in an end-to-end manner guided by the LSM. The source code of our model is publicly available athttps://github.com/illumina7e/TCC-LSTM-LSM. Note that while the proposed model is applied in traffic flow forecasting in this study, it can be easily extended to other temporal forecasting tasks at isolated points, such as the prediction of computer network traffics, electronic demand, outpatients in a hospital, and demand in a shared bicycle or taxi system. We summarize the major contributions of this work as follows.

- We propose a novel LSTM equipped with temporal-aware convolutional context (TCC) blocks and a new loss-switch mechanism (LSM) for traffic flow forecasting at isolated points.
- We propose cascaded temporal-aware convolutional context (TCC) blocks to enlarge the receptive field in temporal contexts to capture much more distinguishable temporal features and a new loss-switch mechanism (LSM) to reduce the effects of non-Gaussian disturbances.
- We evaluate our network on two benchmark sets and compare it with several state-of-the-art models for short-term traffic flow forecasting. Results show our network outperforms previous models in terms of both mean absolute percentage error and root mean square error.

2. Methodology

Fig. 1 presents the workflow of the overall TCC-LSTM-LSM network. Our network take the historical traffic flow sequence as input and outputs the traffic flow prediction in an end-to-end manner. First, it begins by using an initial 1×1 convolution on one day's traffic flow sequence. Second, we employ five cascaded TCC blocks to extract different scales of temporal-aware features to enlarge the receptive field. Third, we employ two LSTM blocks to learn the intrinsic trend and seasonality of the TCC features. Lastly, the traffic flow is predicted by a full connected layer. Furthermore, the whole network is trained by a LSM. Both well-padded wavelet transformed data and the raw data are used to train the network by a loss-switch supervising mechanism, which automatically switches the mean square loss and generalized correntropy induced metric loss during the training process.

In the following subsection, we firstly model the task of traffic flow forecasting, then elaborate how the TCC blocks extract enlarged temporal-aware features. Consequently, we present the generalized correntropy induced metric loss in detail, and then introduce the loss-switch mechanism to train the network for traffic flow forecasting.

2.1. Traffic Flow Modeling

We formulate the STTFF problem at isolated points as:

$$\max_{\hat{f}_{t+\tau}} p\left(\hat{f}_{t+\tau}|f_t f_{t-1}\ldots f_1\right), \tau = 1, \ldots, T, \tag{1}$$

where f_t is traffic flow[1] in time interval t. The goal of STTFF at isolated points is to find optimal prediction $\hat{f}_{t+\tau}$ by maximizing the probability $p(\cdot)$. A common way to address this issue is to train a model guided by an expected loss for certain distributions by minimizing the distance between the ground truth and the prediction.

2.2. Temporal-aware convolutional context blocks

We propose a novel temporal-aware convolutional context (TCC) blocks in order to capture the complicated and subtle trend and seasonality featured inside the traffic flow sequence at isolated points, We first present the architecture of the TCC blocks, and then elaborate how we cascade the TCC blocks to harvest the temporal attention features from different scales in an enlarged receptive field. The architecture of the TCC blocks is shown at the right of

[1] f_t can be either space mean speed or traffic density since they can be derived from each other to represent the situation of traffic flow.

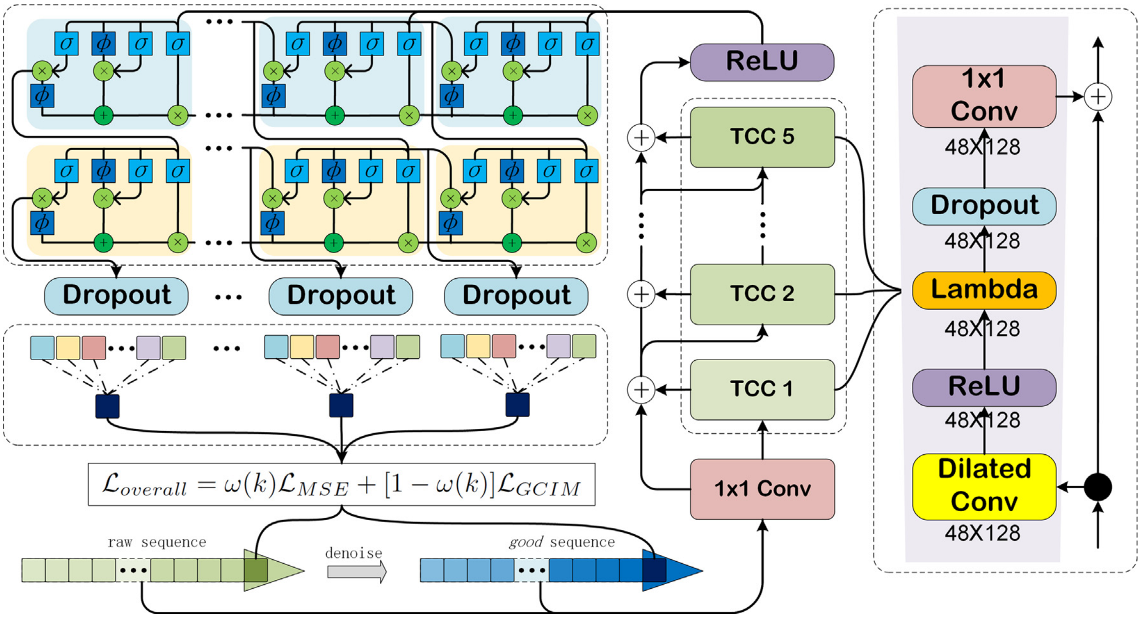

Fig. 1. The schematic illustration of the overall TCC-LSTM-LSM network. (i) we employ an initial 1 × 1 convolution for the raw/*good* input; (ii) we extract temporal-aware features in different receptive field by five cascaded TCC blocks; (iii) we combine the TCC blocks with residual connections for more robust learning; (iv) we concatenate TCC features with LSTM blocks to learn the intrinsic trend and seasonality of the TCC features by a loss-switch supervising mechanism; (v) lastly, we forecast the final result by a full connected layer. The schematic illustration of our *temporal-aware convolutional context (TCC) block* is detailed on the right of the figure. We compute the temporal-aware context by adopting a dilated convolution to enlarge the receptive field, and then a ReLU layer non-linearly maps the features. Then, the results are regularized by the lambda layer. We employ a dropout layer to increase the diversity. The results are concatenate with a residual connection after an 1 × 1 convolution.

Fig. 1. It includes a dilated convolutional layer, a ReLU non-linear activation layer, a lambda layer, a temporal dropout layer, and an 1 × 1 convolutional layer. A residual connection is employed to fuse the outputs of the TCC blocks and the input.

Traditional causal convolution is only able to perceive the historical temporal data with linear lag as increasing the depth of the network. In such a case, if the reception of historical traffic flow sequence is long, which is common in STTFF at isolated points, it requires to train an extremely deep network, which is difficult, if not impossible to carry out, particularly with limited training samples. We address this problem by employing the dilated convolution [37] for temporal data. Denote $\mathbf{x} \in R^n$ as 1D temporal feature sequence and a temporal filter $g : \{0, \ldots, k-1\} \to \mathbb{R}$, we perform the dilated convolution operation $\mathcal{D}$ on the s-th element of the temporal sequence by:

$$\mathcal{D}(s) = (\mathbf{x} *_d g)(s) = \sum_{i=0}^{k-1} g(i) \cdot \mathbf{x}_{s-d \cdot i}, \tag{2}$$

where d is the dilation factor, and k is the filter size. The dilated factor effectively enlarges receptive field of the traditional causal convolution by introducing a stride in the temporal feature sequence. Larger filter size k or dilated factor d produce larger receptive field. Note that when $d = 1$, the dilated convolution degenerates to the causal convolution.

A ReLU layer follows the dilated convolution, and then a successive lambda layer regularizes the results by linear operation. We employ a temporal dropout strategy to encourage diversity of the learned features. Instead of dropping the individual elements, we zero out the entire temporal-aware feature, since the temporal features in the early convolutional layers are highly correlated. In this regard, the temporal dropout strategy empowers the network to learn more independent features.

In order to capture more distinguishable features, we further enlarge the receptive field by cascading more TCC blocks. We first employ a residual connection from the input of a TCC block to its output to stabilize the deeper temporal-aware attention mechanism. This allows the TCC block to learn modifications to the identity mapping instead of the entire transformation. Thus, an 1 × 1 convolution is attached to the temporal dropout layer to ensure the consistent dimensions between the input of the TCC block and the output of the temporal dropout. We then cascade the TCC blocks to distil multiple scales of temporal features. As shown in Fig. 1, after we perform a 1 × 1 initial convolution on the input traffic flow sequence, we concatenate the results with the first TCC block, and then the second TCC block, and so forth. Finally, we fuse the results of these cascaded TCC blocks with residual connections to avoid degradation by transforming the identity mapping to zero mapping. In our implementation, we cascaded five TCC blocks.

2.3. Loss-switch mechanism & training strategy

The features learned from the TCC blocks are then fed into the LSTM network for prediction. Mean square error loss have been widely employed by LSTM network to quantifying how similar the ground truth and the prediction of the network are. The success of such networks with mean square error loss heavily relies on Gaussian and linear assumptions. However, traffic flow is easily affected by unexpected accidents, weather, public events or failure of detectors, which brings impulsive noises or outliers into the traffic flow sequence.

Traffic flow may disturbed with the mixture of two types of non-Gaussian disturbances: light-tailed and heavy-tailed. A high-order statistical metric is more suitable for the light-tailed disturbance, while a lower-order statistic is desired for heavy-tailed impulsive disturbance, such as unexpected accidents. Recently, Chen et al. [38] present a generalized correntropy criterion for a general similarity measurement between two vectors in non-Gaussian cases, which is able to balance the robustness for both disturbances.

The generalized correntropy is based on the well-known generalized Gaussian density:

$$\mathcal{G}_{\alpha,\beta}(e) = \frac{\alpha}{2\beta\Gamma\left(\frac{1}{\alpha}\right)}\exp\left(-\left|\frac{e}{\beta}\right|^{\alpha}\right), \tag{3}$$

where $\Gamma(\cdot)$ is the gamma function. $\alpha, \beta > 0$ are the shape and bandwidth parameters separately. In practice, a finite number of samples $\left\{(x_i, y_i)_{i=1}^{N} | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\right\}$ are drawn from an unknown joint distribution.

$$\widehat{V}_{\alpha,\beta}(\mathcal{X},\mathcal{Y}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{G}_{\alpha,\beta}(x_i, y_i), \tag{4}$$

where $\mathcal{G}_{\alpha,\beta}(\cdot)$ is the generalized Gaussian density.

A comprehensive interpretation of the generalized correntropy is the affine linear function of the α-order absolute moment by dragging the Taylor's expansion on it.

$$\widehat{V}_{\alpha,\beta}(\mathcal{X},\mathcal{Y}) = \frac{\alpha}{2\beta\Gamma\left(\frac{1}{\alpha}\right)}\sum_{p=0}^{\infty}(-1)^{p}\frac{1}{p!|\beta|^{\alpha p}}\mathbf{E}\left[|\mathcal{X}-\mathcal{Y}|^{\alpha p}\right], \tag{5}$$

where $\mathbf{E}$ is the mathematical expectation. The generalized correntropy includes all moment statistical information, where the high-order statistical metric is suitable for the light-tailed disturbances, while the lower-order statistic is desired for heavy-tailed impulsive disturbances in the traffic flow.

We integrate a novel loss derived from the generalized correntropy, termed generalized correntropy induced metric (GCIM) loss [38], as a metric for our network.

$$\mathcal{L}_{GCIM} = \mathcal{G}_{\alpha,\beta}(0) - \widehat{V}_{\alpha,\beta}(\mathcal{X},\mathcal{Y}). \tag{6}$$

Different from MSE loss, which increases quadratically to large error outliers, GCIM loss approaches the mean α-power error loss, when the errors are relatively small, and close to a constant for large outliers. This is why GCIM insensitive to large outliers, which often occur in traffic flow sequences. However, the parameters of our network are initialized randomly, since our network is quite different from the well-trained models on shelf. Thus, the training error will be relatively large for the initial epoches. As discussed, the GCIM loss is insensitive to large errors. This property make it ineffective to learn efficiently at the initial stage, as the network lacks of strong enough explicitly encouragement. To efficiently learn a strong one, we present a loss-switch mechanism to optimize the whole network in the training process.

In detail, the overall loss of our network is defined as:

$$\mathcal{L}_{overall} = \omega(k)\mathcal{L}_{MSE} + [1-\omega(k)]\mathcal{L}_{GCIM}, \tag{7}$$

where $\omega(k) = \frac{1}{1+e^{\mathcal{M}(k-k_0-\varepsilon)}}$, $\mathcal{L}_{MSE}$ is the MSE loss, k is index of the epoches, k_0 is the epoches using *good* traffic flow sequence, and $\mathcal{M}$ is a big integer, which is set to 100 in this work. ε is set to 0.1 is this work.

We train the aforementional network by the following steps. First, we construct a small training set of *good* traffic flow sequence by padding the missing data in the raw traffic flow sequence with the filling method proposed by [39]. Second, the padded training data are denoised by wavelet transform with Daubechies 4 mother-wavelet employed in [4]. The *good* small training dateset is copied and pasted repeatedly to construct a *good* training dataset, whose size is equal to the raw training dataset. Third, we combine each sequence in the *good* training dataset and the raw dataset by $\omega(k)x_i^{(good)} + [1-\omega(k)]x_i^{(raw)}$ for each input. When $k \leqslant k_0, \omega(k) = 1$ and $[1-\omega(k)] = 0$ considering the round-off, whereas $k > k_0, \omega(k) = 0$ and $[1-\omega(k)] = 1$. When $k \leqslant k_0$, we select the *good* traffic flow sequences and the MSE loss for training. Otherwise, we select the raw traffic flow sequences and the GCIM loss for training. In this way, we can hot-switch the training dataset and loss function without interrupting the training process. Hence, we term this fashion as loss-switch mechanism, which can be further generalized and is promising for other learning tasks with noisy dataset.

3. Experiments and discussions

3.1. Benchmark datasets

Two typical benchmark datasets are employed to validate the proposed method, which are widely used in traffic flow forecasting [40,9–12,4–6]. The first benchmark dataset is from the traffic data acquisition and distribution system (TDAD).[2] Four traffic flow samples are collected from June 6, 2005 to July 10, 2005 at four isolated points. The second benchmark dataset is collected from the largest publicly available database, Caltrans Performance Measurement System (PeMS) [3]. The data are collected every 30 s from 39,000 individual detectors, which span the freeway system across all major metropolitan areas of the state of California. From 39,000 points, we collect traffic flow samples from six boundary points during January 1, 2018 to July 8, 2018. The standard time interval of the first dataset is 10 min, while time interval of the second one is 5 min. We train our model separately on each point and 80% of the data are used for training while the rest data are used for testing.

3.2. Evaluation criteria

We employ two criteria to quantitatively evaluate the traffic flow forecasting performance, which have been widely used [9–12]. The first one is the mean absolute percentage error (MAPE) and the other is the root mean square error (RMSE):

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{\hat{f}_t - f_t}{f_t}\right|, \tag{8}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\hat{f}_t - f_t\right)^2}, \tag{9}$$

where f_t is the observed traffic flow at time interval t, and $\hat{f}_t$ is the predicted traffic flow.

3.3. Experimental settings

3.3.1. Shape & bandwidth parameters

We empirically determine the α and β by using the principles discussed in [38]. The order-α GCIM loss measures the localized similarity like L_α norm to L_0 norm in different regions. When two random variables are relatively small compared with the kernel size β, the GCIM behaves like mean α-power loss, and when two random variables get large, the GCIM evolves mean absolute loss, and the GCIM falls to 0–1 loss as they are far apart as outliers. For example, if we set $\alpha = 4$, the GCIM behaves like least mean fourth algorithm when the error is relatively smaller than the bandwidth, and behaves as least square mean as the error get larger. As discussed in [38], large shape parameters α achieve faster convergence speed and a lower steady-state mean square deviation, while the stability of large α is not guaranteed, which depend on the related quantity of noise, as well as the initial values of the network parameters. The α is often set no larger than 4, since the behaviours of a large one is similar, but the stability is less guaranteed. On the other hand, the kernel size β determines the sensitivity of model to large outliers, e.g. caused by hardware failure. In this case, a lower-order statistical measurement usually gains

[2] http://www.its.washington.edu/tdad/

[3] http://pems.dot.ca.gov/

Table 1
Illustration of how different α and β affect RMSE and MAPE on ES855d of TDAD dataset.

α	β	0.01	0.05	0.14	0.25	0.5
1.5	RMSE	39.91	34.24	33.20	35.05	35.17
	MAPE	15.21%	12.25%	10.82%	11.60%	11.52%
2	RMSE	36.79	33.30	**31.60**	32.54	33.48
	MAPE	13.09%	11.15%	**10.47%**	11.01%	11.25%
2.5	RMSE	37.63	34.81	33.84	34.05	35.28
	MAPE	15.48%	11.72%	12.30%	12.00%	11.97%

Table 2
Demonstration of different d and k for our network on ES708d and ES088d of TDAD dataset

	Filter	size	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
ES708d	Exponential	RMSE	22.78	**21.71**	22.50	22.79	23.21
	Strategy	MAPE	9.02%	**8.32%**	8.59%	8.57%	8.69%
	Linear	RMSE	23.81	23.92	23.53	23.56	25.37
	Strategy	MAPE	9.11%	9.00%	8.97%	8.66%	9.47%
ES088d	Exponential	RMSE	39.94	**37.38**	39.08	38.58	41.28
	Strategy	MAPE	8.02%	**7.45%**	7.83%	7.49%	8.17%
	Linear	RMSE	41.86	40.41	39.34	40.45	43.42
	Strategy	MAPE	7.91%	7.87%	7.78%	7.82%	8.33%

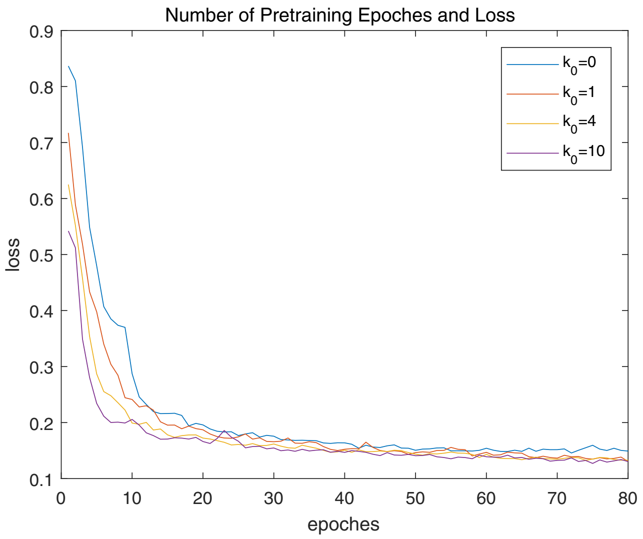

Fig. 2. The network loss of the initial 80 epoches with $k_0 = 0, 1, 4, 10$.

more robustness. For example, the mean absolute value loss is more robust to large outliers. A rational choice is set β to the same quantity of the error. Since the prediction and the groundtruth is normalized, we recommend to set $\beta < 1$. We illustrate how α and β affect RMSE and MAPE in Table 1 by taking ES855d of TDAD dataset as an example.

3.3.2. The dilated factor & filter size

We test two strategies to increase the dilated factor, namely exponential strategy and linear strategy. In exponential strategy, we set the dilated factor $d = 1, 2, 4, 8, 16$ in TCC 1, 2, 3, 4, and 5, respectively, while in linear strategy, we set the dilated factor $d = 1, 2, 3, 4, 5$. In each strategy, we test the filter size k from 1 to 5. Table 1 demonstrate the RMSE and MAPE of both strategies on ES708d and ES088d of TDAD dataset.

From Table 2, we can easily observe that the exponential strategy outperforms the linear strategy for our network. This is because the exponential strategy enlarges the receptive field of the proposed network. The filter size k has weaker impact on the accuracy than the dilated factor d. When $k = 1$, the dilated convolution degenerates to the causal convolution. When $k \geqslant 2$, the dilated convolution involves the adjacency. Smaller filter size take a higher resolution on the responding features, which may be easily affected by outliers. Larger filter size fuse the feature with low resolution, that may miss some important clues. Overall, the filter size is more robust than the dilated factor.

Table 3
Comparisons with state-of-the-art methods on TDAD dataset.

TDAD		ES088d			ES645d			ES708d			ES855d		
dataset		10 min	20 min	30 min	10 min	20 min	30 min	10 min	20 min	30 min	10 min	20 min	30 min
ANN	RMSE	43.11	47.92	49.93	27.19	32.76	35.19	26.66	29.09	31.53	36.43	44.65	46.37
	MAPE	8.68%	9.80%	10.01%	11.51%	13.74%	14.71%	12.73%	15.20%	16.30%	13.68%	19.14%	18.96%
DBN	RMSE	39.86	46.91	48.24	26.71	30.28	34.18	24.11	27.23	30.97	35.17	41.41	44.46
	MAPE	8.21%	10.45%	10.41%	10.87%	13.76%	16.65%	12.16%	15.25%	18.27%	13.55%	18.22%	20.25%
SAE	RMSE	39.39	46.54	47.43	25.60	29.68	32.59	24.64	27.01	28.95	35.20	40.56	43.07
	MAPE	7.78%	9.63%	9.73%	9.78%	12.50%	13.42%	12.23%	13.74%	15.03%	13.42%	18.11%	18.80%
TCN	RMSE	42.58	45.34	47.72	27.00	28.05	31.20	25.71	26.44	28.72	35.42	39.22	42.61
	MAPE	9.90%	9.10%	9.69%	9.82%	10.50%	11.82%	12.50%	11.20%	12.67%	13.56%	13.98%	15.28%
LSTM	RMSE	42.75	45.56	47.56	25.78	27.91	31.01	25.42	26.63	27.81	34.27	37.57	41.82
	MAPE	9.06%	9.73%	10.13%	9.70%	10.47%	11.83%	11.02%	11.88%	12.21%	13.04%	14.01%	16.75%
TCC-LSTM	RMSE	38.53	43.30	47.58	25.39	27.21	29.69	22.97	25.48	26.62	33.06	36.00	38.18
-MSE	MAPE	7.77%	8.36%	9.14%	9.16%	9.95%	10.70%	8.85%	9.63%	10.12%	11.03%	12.63%	12.87%
TCC-LSTM	RMSE	**37.39**	**40.53**	**45.25**	**24.98**	**26.16**	**28.21**	**21.71**	**24.14**	**25.03**	**31.65**	**34.34**	**36.70**
-LSM	MAPE	**7.45%**	**7.97%**	**8.52%**	**8.90%**	**9.61%**	**10.21%**	**8.32%**	**9.45%**	**9.23%**	**10.47%**	**11.34%**	**12.09%**

Table 4
Comparisons with state-of-the-art methods on PeMS dataset.

PeMS		1108299			1108380			1108439			1108599			1114254			1117857		
dataset		5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min
ANN	RMSE	18.59	18.90	19.25	25.47	27.10	28.74	41.98	45.50	47.29	25.09	27.70	27.21	43.28	46.24	48.10	33.75	34.61	36.34
	MAPE	13.84%	25.50%	42.50%	11.56%	11.66%	11.82%	7.48%	8.26%	8.33%	10.13%	11.06%	10.92%	9.40%	9.85%	10.27%	10.07%	10.73%	11.20%
DBN	RMSE	17.97	18.26	19.35	24.60	26.23	28.10	41.31	44.21	46.70	24.07	25.88	26.84	40.70	44.38	47.86	33.00	33.34	34.81
	MAPE	13.05%	14.31%	13.89%	10.11%	10.53%	11.24%	7.40%	7.97%	8.37%	9.43%	9.80%	10.24%	8.39%	8.95%	9.58%	9.86%	9.94%	10.43%
SAE	RMSE	17.43	18.17	18.55	23.36	25.17	27.68	39.56	42.75	45.49	23.90	25.18	26.01	40.82	43.12	45.76	30.71	32.54	34.73
	MAPE	12.84%	14.79%	15.16%	10.18%	10.41%	13.93%	7.15%	7.46%	7.89%	9.57%	9.94%	9.82%	8.97%	8.67%	9.28%	9.76%	9.95%	10.82%
TCN	RMSE	18.33	18.45	19.97	24.53	26.79	28.24	40.20	42.07	45.38	23.82	25.22	25.63	41.37	43.04	46.61	31.16	33.53	34.34
	MAPE	12.37%	13.30%	15.09%	10.33%	12.99%	12.18%	7.18%	7.53%	9.81%	9.44%	10.53%	9.79%	8.55%	9.04%	9.28%	9.09%	9.43%	11.56%
LSTM	RMSE	17.27	18.41	18.59	23.87	25.77	27.56	39.54	42.62	44.76	23.93	25.85	25.96	40.37	42.98	45.93	31.05	34.00	34.77
	MAPE	13.44%	15.54%	15.89%	10.06%	10.67%	11.34%	7.29%	7.95%	8.04%	9.22%	9.95%	10.04%	8.89%	8.71%	9.64%	9.68%	11.00%	11.63%
TCC-LSTM	RMSE	17.12	17.71	18.73	22.63	24.83	26.72	39.52	40.52	42.07	24.01	24.95	25.18	40.34	42.37	44.68	30.95	31.99	32.68
-MSE	MAPE	12.10%	13.02%	13.09%	9.30%	9.79%	10.12%	7.22%	7.01%	7.48%	9.25%	9.31%	9.67%	8.01%	8.54%	9.23%	9.22%	9.40%	9.66%
TCC-LSTM	RMSE	**16.62**	**17.12**	**18.06**	**22.38**	**23.83**	**25.37**	**38.55**	**39.88**	**41.14**	**23.17**	**24.06**	**24.44**	**39.18**	**42.06**	**44.40**	**29.78**	**31.16**	**32.10**
-LSM	MAPE	**11.55%**	**12.93%**	**12.92%**	**8.99%**	**9.17%**	**9.53%**	**6.56%**	**6.86%**	**6.95%**	**8.65%**	**8.87%**	**8.92%**	**7.78%**	**8.13%**	**8.47%**	**8.63%**	**9.12%**	**9.18%**

Table 5
Comparisons with two state-of-the-art spatio-temporal models on TDAD dataset.

TDAD		ES088d			ES645d			ES708d			ES855d		
dataset		10 min	20 min	30 min	10 min	20 min	30 min	10 min	20 min	30 min	10 min	20 min	30 min
GraphWaveNet [43]	RMSE	40.45	43.37	46.42	26.38	27.81	29.51	24.13	25.65	27.35	33.95	37.32	40.13
	MAPE	8.98%	8.84%	9.21%	9.66%	10.36%	10.72%	10.91%	10.73%	11.35%	12.42%	13.16%	14.29%
GMAN [44]	RMSE	39.48	42.53	46.48	25.69	26.5	29.73	23.23	24.71	26.32	33.1	36.22	39.14
	MAPE	8.28%	8.32%	9.15%	9.06%	9.86%	10.82%	9.51%	9.93%	10.25%	11.52%	12.16%	12.89%
TCC-LSTM-LSM	RMSE	**37.39**	**40.53**	**45.25**	**24.98**	**26.16**	**28.21**	**21.71**	**24.14**	**25.03**	**31.65**	**34.34**	**36.7**
	MAPE	**7.45%**	**7.97%**	**8.52%**	**8.90%**	**9.61%**	**10.21%**	**8.32%**	**9.45%**	**9.23%**	**10.47%**	**11.34%**	**12.09%**

3.3.3. Number of pretraining epoches

We employ a loss-switch strategy in the training stage. A small portion of training data are preprocessed to produce *high quality* traffic flow sequence. The network are pretrained on these *high quality* data to initialize the parameters. Fig. 2 demonstrates the loss of first 80 epoches of ES855d of TDAD training for our TCC-LSTM-LSM. If $k_0 = 0$, that means cancellation of pretraining. In Fig. 2, we can observe that the blue line decreases more slowly than the others, and the convergence speed is obviously improved even after one epoch pretraining. After 1, 4, or 10 epoch(es) pretraining, the loss decreases faster. Initially, the network loss is relatively large, but the GCIM is insensitive to large error. Thus, the network converges slowly. The MSE is a global metric, which functions well when the error large is large. After at least one epoch pretraining, the parameters of the network converges to a better ones. However, since only a small portion of training data take part in this stage, the pretraining stage can not last too many epoches to prevent overfitting. In our experiment, we set k_0 at most 10.

3.4. Performance comparisons

We compare our method with five start-of-the-art temporal methods: artificial neural network (ANN) [41], deep belief network (DBN) [9], stacked autoencoder (SAE) [10], temporal convolutional network (TCN) [42], long short term memory (LSTM) [14]. The number of layer of the ANN is 6, the number of the nodes in the hidden layers is 128, the activation function is *ReLU*. Each restricted Boltzmann machine (RBM) of the DBN uses *ReLU* activation function with a learning rate of 0.05, and the training epoch number is 10. The whole DBN is structured by stacking two layers of RBMs. Each layer contains 256 RBMs. The whole DBN is trained 100 epochs with a dropout rate of 0.2 and a learning rate of 0.1. The architecture of the SAE is $[400, 200, 100]$, the learning rate is 0.1, the weight regularization penalty is set to 0.05, the dropout is set to 0.2, and the sparsity is set to 0.03. The layers of TCN are connected hierarchically with a dilation factor of 2 and the kernel size is 3. The exponential dilation is used in the TCN. The model is optimized by Adam with an initial learning rate of 0.002. The number of hidden units of the LSTM model is 64, the activation function is *tanh* for the LSTM layer and *sigmoid* for the dense layer with a learning rate of 0.01. The number of hidden units is $[32, 32]$, and the batch size is 32. Our model is optimized by Adam with an initial learning rate of 0.1 and a decay rate of $1e-6$. The default epoch number is set to 100. We apply early stopping to stop the training process when convergence. In each epoch, the batch size to input is set to 64, and the dropout rate is set to 0.2.

Tables 3 and 4 report the comparison results of ten isolated points for the next three time interval in two benchmark datasets, where we can easily observe that deep neural networks (DBN, SAE, TCN, LSTM, and our methods) perform much better than the shallow network (ANN), demonstrating the deep networks are capable in capturing complicated and subtle patterns in traffic flow. It is observed that our method outperforms our rivals in terms of both RMSE and MAPE at the 10 isolated points in both benchmark datasets. Logically, our method outperforms TCN and LSTM for three reasons: 1) our TCC blocks are capable of harvesting the abstract temporal attention features in an enlarged receptive field; 2) the proposed LSM-guided LSTM network is able to catch up the change of the traffic flow for the most recent time interval; and 3) the proposed LSM is robust to large impulsive disturbances caused by unexpected incident or other outliers. We also compare our network (TCC-LSTM-LSM) with our network only equipped with MSE loss (TCC-LSTM-MSE). TCC-LSTM-LSM outperforms TCC-LSTM-MSE in all cases, which demonstrate that the traffic flow is easily disturbed by impulsive noises and the GCIM is robust to this kind of disturbances, which empowers our network to outperform the version with only MSE.

Table 6
Comparisons with two state-of-the-art spatio-temporal models on PeMS dataset.

PeMS		1108299			1108380			1108439			1108599			1114254			1117857		
dataset		5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min	5 min	10 min	15 min
GraphWaveNet [43]	RMSE	17.4025	17.6675	19.1425	23.4925	25.55	26.6725	39.1375	41.3225	43.37	23.4075	24.78	25.1325	40.6225	42.495	45.6575	30.515	32.6875	33.68
	MAPE	12.12%	13.21%	14.40%	9.95%	11.74%	11.27%	7.03%	7.34%	8.60%	9.24%	10.02%	9.47%	8.36%	8.81%	9.08%	9.03%	9.40%	10.97%
GMAN [44]	RMSE	17.075	17.485	18.515	23.055	24.71	25.905	39.075	40.575	42.16	23.395	24.54	25.135	40.275	42.55	45.105	30.47	31.845	33.22
	MAPE	11.86%	13.12%	13.71%	9.56%	10.48%	10.36%	6.87%	7.15%	7.38%	9.05%	9.50%	9.16%	8.17%	8.59%	8.88%	8.96%	9.38%	10.37%
TCC-LSTM-LSM	RMSE	**16.62**	**17.12**	**18.06**	**22.38**	**23.83**	**25.37**	**38.55**	**39.88**	**41.14**	**23.17**	**24.06**	**24.44**	**39.18**	**42.06**	**44.4**	**29.78**	**31.16**	**32.1**
	MAPE	**11.55%**	**12.93%**	**12.92%**	**8.99%**	**9.17%**	**9.53%**	**6.56%**	**6.86%**	**6.95%**	**8.65%**	**8.87%**	**8.92%**	**7.78%**	**8.13%**	**8.47%**	**8.63%**	**9.12%**	**9.18%**

Table 7
The comparison of computation time between the TCC-LSTM-LSM model and two state-of-the-art spatio-temporal models.

Time (seconds)	Train	Test
Graph WaveNet	109.32	3.9
GMAN	130.57	5.6
TCC-LSTM-LSM	157.27	6.2

To demonstrate our model can more effectively capture the temporal dependency in the traffic flow sequence, we also take an *unfair* comparison with two state-of-the-art spatio-temporal models, e.g. the Graph WaveNet [43] and the GMAN [44], on two benchmark datasets. The Graph WaveNet is proposed by Wu et al. [43]. In this paper, a novel graph neural network architecture is presented by introducing a adaptive dependency matrix and a node embedding mechanism. The graph multi-attention network (GMAN) is proposed by Zheng et al. [44], which adapts a encoder-decoder architecture consist of multiple spatio-temporal attention blocks to model the impact of the spatio-temporal factors. A transform attention layer between the encoder and the decoder models the relationship between the historical and future time steps. Our model is trained independently on each point of the TDAD dataset and the PeMS dataset without geographic locations, while the counterparts are trained on all points with geographic information simultaneously, so they can leverage the spatial dependency of the traffic flow from all the points. The Graph WaveNet is eight layers with a sequence of dilation factors, e.g. $[1, 2, 1, 2, 1, 2, 1, 2]$. The graph convolution layer with diffusion steps $K = 2$. The initial learning rate of the Adam optimizer is 0.001, and the dropout of the output of the graph convolution layer is 0.3. The GMAN is trained by Adam optimizer with an initial learning rate of 0.001. The number of traffic conditions is set to 1. The number of ST-Attention blocks is set to 3, the number of attention heads is set to 8, and the dimension of each attention head is set to 8. The results are reported in Table 5 and Table 6. Although the Graph WaveNet and GMAN takes the advantages of the spatio-temporal dependency of traffic flow from all the points, our method still outperforms them for 5, 10, 15, 20, and 30 min forecasting on both datasets. Such results demonstrate our model can more effectively capture the temporal dependency in the traffic flow sequence. There may be two reasons. First, the complex spatio-temporal relationships *are encoded* inside the big data of every day's traffic flow. Second, the social circumstances near the road segment is difficult, if not impossible to be quantitatively defined [45]. For example, two road segments have the same number of lanes, the same slope of the pavement, and the same length, but the economics of the areas or the demographic density near the road segments may vary from each other, which subsequently affects the traffic flow. Instead of predefined an explicit graph structure, developing a new deep learning network that can automatically and adaptively learn the spatio-temporal relationship from the temporal sequences would be an interesting future work. The computation time is also reported in Table 7. The computation time of the Graph WaveNet and the GMAN is the total time computed on all the points of the PeMS dataset, while the computation time of the proposed model is the sum of the time computed on each point. Our model is slightly slower than the counterparts, since it is trained on each points.

We further provide comparison results visually in Fig. 3, which show various challenging cases, e.g., the traffic flow changes dramatically during peak hours. Without understanding the global trend and seasonality, as well as inconspicuous clues of the traffic flow, it is hard to follow up such changes by the other models, so relatively smooth lines are predicted by these models. From the

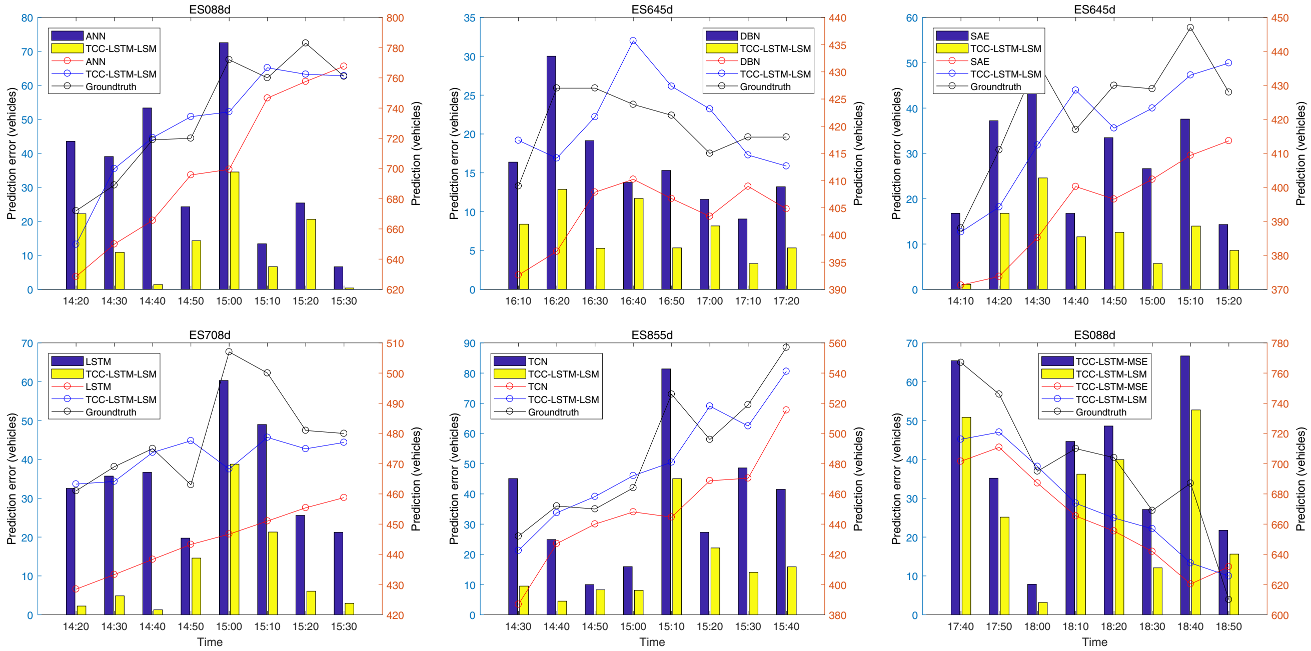

Fig. 3. Six examples demonstrate our network outperforms the state-of-the-art models during peak hours.

results, we can see that our network can effectively catch up the traffic flow and produce more accurate forecasting.

4. Conclusions

This paper presents a novel long short-term memory network with temporal-aware convolutional context and a new loss-switch mechanism. Our key ideas are to, i) drill multi-scale temporal-aware features to enlarge the receptive field by cascading TCC blocks; ii) formulate a temporal-aware attention mechanism to learn intricate temporal features; and iii) learn an outlier insensitive network equipped a loss-switch mechanism embedded with mean square error loss and the generalized correntropy induced metric. We apply our network for traffic flow forecasting at isolated points and extensive experiments from two common benchmark datasets well demonstrate the effectiveness and robustness of the proposed methods by comparing with the state-of-the-art traffic flow forecasting methods.

In future, we plan to explore the potential of our network for other time series forecasting applications, such as sharing bicycle demand forecasting, energy consumption forecasting, or railway passenger forecasting.

CRediT authorship contribution statement

Huakang Lu: Methodology, Formal analysis, Investigation, Writing - review & editing. **Zuhao Ge:** Formal analysis, Resources. **Youyi Song:** Formal analysis, Data curation, Writing - original draft. **Dazhi Jiang:** Formal analysis, Funding acquisition. **Teng Zhou:** Methodology, Software, Supervision. **Jing Qin:** Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the NSFC (No. 61902232), the Natural Science Foundation of Guangdong Province (No. 2018A030313291), the Education Science Planning Project of Guangdong Province (2018GXJK048), the Guangdong Special Cultivation Funds for College Students' Scientific and Technological Innovation (No. pdjh2020b0222), the grant from the Hong Kong Polytechnic University (No. 1ZE8J), the STU Scientific Research Foundation for Talents (NTF18006), and the 2020 Li Ka Shing Foundation Cross-Disciplinary Research Grant (No. 2020LKSFG05D, 2020LKSFG04D).

References

[1] K. Boriboonsomsin, M.J. Barth, W. Zhu, A. Vu, Eco-routing navigation system based on multisource historical and real-time traffic information, IEEE Trans. Intell. Transp. Syst. 13 (4) (2012) 1694–1704.

[2] A. Di Febbraro, D. Giglio, N. Sacco, A deterministic and stochastic petri net model for traffic-responsive signaling control in urban areas, IEEE Trans. Intell. Transp. Syst. 17 (2) (2016) 510–524.

[3] M. Dotoli, H. Zgaya, C. Russo, S. Hammadi, A multi-agent advanced traveler information system for optimal trip planning in a co-modal framework, IEEE Trans. Intell. Transp. Syst. 18 (9) (2017) 2397–2412.

[4] T. Zhou, D. Jiang, Z. Lin, G. Han, X. Xu, J. Qin, Hybrid dual kalman filtering model for short-term traffic flow forecasting, IET Intel. Transport Syst. (2019) 1–10, https://doi.org/10.1049/iet-its.2018.5385.

[5] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, J. Qin, A noise-immune kalman filter for short-term traffic flow forecasting, Physica A 536 (2019) 122601.

[6] S. Zhang, Y. Song, D. Jiang, T. Zhou, J. Qin, Noise-identified kalman filter for short-term traffic flow forecasting, in: The 15th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2019), IEEE, 2019, pp. 1–5.

[7] M.-C. Tan, S.C. Wong, J.-M. Xu, Z.-R. Guan, P. Zhang, An aggregation approach to short-term traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 10 (1) (2009) 60–69.

[8] M. Lippi, M. Bertini, P. Frasconi, Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning, IEEE Trans. Intell. Transp. Syst. 14 (2) (2013) 871–882.

[9] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: Deep belief networks with multitask learning, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.

[10] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, et al., Traffic flow prediction with big data: A deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 865–873.

[11] T. Zhou, G. Han, X. Xu, Z. Lin, C. Han, Y. Huang, J. Qin, δ)agree adaboost stacked autoencoder for short-term traffic flow forecasting, Neurocomputing 247 (2017) 31–38.

[12] T. Zhou, G. Han, X. Xu, C. Han, Y. Huang, J. Qin, A learning-based multimodel integrated framework for dynamic traffic flow forecasting, Neural Process. Lett. (2018) 1–24.

[13] J. Mackenzie, J.F. Roddick, R. Zito, An evaluation of htm and lstm for short-term arterial traffic flow prediction, IEEE Trans. Intell. Transp. Syst. 99 (2018) 1–11.

[14] B. Yang, S. Sun, J. Li, X. Lin, Y. Tian, Traffic flow prediction using lstm with feature enhancement, Neurocomputing (2018) 320–327.

[15] Y. Zhang, S. Wang, B. Chen, J. Cao, Z. Huang, Trafficgan: Network-scale deep traffic prediction with generative adversarial nets, IEEE Trans. Intell. Transp. Syst.

[16] H.-F. Yang, T.S. Dillon, E. Chang, Y.-P.P. Chen, Optimized configuration of exponential smoothing and extreme learning machine for traffic flow forecasting, IEEE Trans. Industr. Inf. 15 (1) (2018) 23–34.

[17] W. Cai, J. Yang, Y. Yu, Y. Song, T. Zhou, J. Qin, Pso-elm: A hybrid learning model for short-term traffic flow forecasting, IEEE Access (2020) 1–10.

[18] Z. Li, Z. Zheng, S. Washington, Short-term traffic flow forecasting: a component-wise gradient boosting approach with hierarchical reconciliation, IEEE Trans. Intell. Transp. Syst.

[19] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, X. Zhou, Lc-rnn: A deep learning model for traffic speed prediction, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), 2018, pp. 3470–3476.

[20] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), 2018, pp. 3634–3640.

[21] C. Huang, C. Zhang, J. Zhao, X. Wu, D. Yin, N. Chawla, Mist: A multiview and multimodal spatial-temporal learning framework for citywide abnormal event forecasting, The World Wide Web Conference, 2019, pp. 717–728.

[22] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, J. Zhang, Urban traffic prediction from spatio-temporal data using deep meta learning, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1720–1730.

[23] Z. Pan, W. Zhang, Y. Liang, W. Zhang, Y. Yu, J. Zhang, Y. Zheng, Spatio-temporal meta learning for urban traffic prediction, IEEE Trans. Knowl. Data Eng.

[24] X. Wu, C. Huang, C. Zhang, N.V. Chawla, Hierarchically structured transformer networks for fine-grained spatial event forecasting, Proceedings of The Web Conference 2020, 2020, pp. 2320–2330.

[25] Y. Liang, K. Ouyang, Y. Wang, Y. Liu, J. Zhang, Y. Zheng, D.S. Rosenblum, Revisiting convolutional neural networks for citywide crowd flow analytics, Proceedings of ECML-PKDD, 2020.

[26] L. Cai, Y. Yu, S. Zhang, Y. Song, Z. Xiong, T. Zhou, A sample-rebalanced outlier-rejected k)nearest neighbor regression model for short-term traffic flow forecasting, IEEE Access 8 (2020) 22686–22696.

[27] Z. Pan, Z. Wang, W. Wang, Y. Yu, J. Zhang, Y. Zheng, Matrix factorization for spatio-temporal neural networks with applications to urban flow prediction, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 2683–2691.

[28] Z. Dowd, A.Y. Franz, J.S. Wasek, A decision-making framework for maintenance and modernization of transportation infrastructure, IEEE Trans. Eng. Manage.

[29] Y.-E. Sun, H. Huang, S. Chen, H. Xu, K. Han, Y. Zhou, Persistent traffic measurement through vehicle-to-infrastructure communications in cyber-physical road systems, IEEE Trans. Mobile Computing.

[30] Y. Wang, J.H. Van Schuppen, J. Vrancken, Prediction of traffic flow at the boundary of a motorway network, IEEE Trans. Intell. Transp. Syst. 15 (1) (2014) 214–227.

[31] Y. Tian, K. Zhang, J. Li, X. Lin, B. Yang, Lstm-based traffic flow prediction with missing data, Neurocomputing 318 (2018) 297–305.

[32] F. Zhao, G.-Q. Zeng, K.-D. Lu, Enlstm-wpeo: Short-term traffic flow prediction by ensemble lstm, nnct weight integration and population extremal optimization, IEEE Transactions on Vehicular Technology.

[33] A. Koesdwiady, R. Soua, F. Karray, Improving traffic flow prediction with weather information in connected cars: A deep learning approach, IEEE Trans. Veh. Technol. 65 (12) (2016) 9508–9517.

[34] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, Y. Zhang, Deep and embedded learning approach for traffic flow prediction in urban informatics, IEEE Trans. Intell. Transp. Syst.

[35] L. Cai, M. Lei, S. Zhang, Y. Yu, T. Zhou, J. Qin, A noise-immune lstm network for short-term traffic flow forecasting, Chaos: Interdisciplinary J. Nonlinear Sci. 30 (2)(2020) 023135.

[36] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, arXiv preprint arXiv:1609.03499.

[37] H. Zhu, Y. Qiao, G. Xu, L. Deng, Y. Yu-Feng, Dspnet: A lightweight dilated convolution neural networks for spectral deconvolution with self-paced learning, IEEE Trans. Ind. Informatics.

[38] B. Chen, L. Xing, H. Zhao, N. Zheng, J.C. Prı, et al., Generalized correntropy for robust adaptive filtering, IEEE Trans. Signal Process. 64 (13) (2016) 3376–3387.

[39] X. Yi, Y. Zheng, J. Zhang, T. Li, St-mvl: filling missing values in geo-sensory time series data, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 2704–2710.

[40] Y. Xie, Y. Zhang, Z. Ye, Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition, Computer-Aided Civil Infrastructure Eng. 22 (5) (2007) 326–334.

[41] D. Chen, Research on traffic flow prediction in the big data environment based on the improved rbf neural network, IEEE Trans. Industr. Inf. 13 (4) (2017) 2000–2008.

[42] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271.

[43] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 1907–1913.

[44] C. Zheng, X. Fan, C. Wang, J. Qi, Gman: A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 1234–1241.

[45] H. Lu, D. Huang, S. Youyi, D. Jiang, T. Zhou, J. Qin, St-trafficnet: A spatial-temporal deep learning network for traffic forecasting, Electronics (2020) 1–17.

Huakang Lu is currently pursuing the bachelor's degree with the Department of Computer Science at Shantou University, China. His research interests include intelligent transportation systems, machine learning and computer vision, etc.

Zuhao Ge is currently pursuing the bachelor's degree with the Department of Computer Science and Technology, College of Engineering, Shantou University, China. His research interests include machine learning and computer vision.

Youyi Song is a Ph.D. candidate with the Centre of Smart Health, Hong Kong Polytechnic University. His research interests include clinical science, medical image segmentation, machine learning, and data analysis.

Dazhi Jiang received his BA in Computer Science from the China University of Geoscience (Wuhan) in 2004. He obtained his PhD from the State Key Laboratory of Software Engineering, Wuhan University, China in 2009. Since then, he has been with the Department of Computer Science, Shantou University, China where he was a Professor. His research interests include affective computing, deep learning, data mining and applications of artificial intelligence.

Teng Zhou is an Assistant Professor with the Department of Computer Science, Shantou University, and also serves as a Research Associate at the Center of Smart Health, the Hong Kong Polytechnic University. His research interests include intelligent transportation system and machine learning.

Jing Qin is an associate professor at the Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University. His research interests are intelligent traffic system, VR-based surgical simulation, multisensory human-computer interaction, and biomechanical modeling. Qin received a Ph.D. from the Chinese University of Hong Kong?s Department of Computer Science and Engineering.