IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 29, 2020, accepted October 16, 2020, date of publication November 9, 2020,
date of current version November 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3036652

Context-Aware Traffic Prediction Framework Based on Series Decomposition

TAO RUAN[1], DEXING WU[1], TIANYI CHEN[2], CANGHONG JIN[2], LEI XU[1], SHENGLI ZHOU[3], AND ZHEN JIANG[4]

[1]Zhejiang Institute of Transportation Company Ltd., Hangzhou 310015, China
[2]Zhejiang University City College, Hangzhou 310015, China
[3]Zhejiang Police College, Hangzhou 310000, China
[4]Zhejiang University, Hangzhou 310012, China

Corresponding author: Tao Ruan (taoruan@outlook.com)

This work was supported in part by the Research Project of Zhejiang Communication Investment Group Company Ltd., under Grant 201902, in part by the Zhejiang Department of Education under Grant Y201941372, in part by the Zhejiang Department of Transportation under Grant 2019006 and Grant 2020007, and in part by the Natural Science Foundation of Zhenjiang Province of China under Grant LY19A010028 and Grant LY21F020003.

ABSTRACT Forecasting traffic flow is a typical time series problem, which has attracted increasing attention due to the urgent need in intelligent transportation systems. Although numerous time series forecasting methods have been investigated in past decades, from statistics based models to deep neural networks models, the main disadvantages of aforementioned work could be summarized as follows: 1) incapable to handle the complexity and uncertainty of series; 2) incapable to consider external features such as spatial information and importance of points during the learning process; 3) unstable performance on forecasting task given various data patterns. In this study, a novel strategy was proposed to extract context-awareness information and then integrated with Temporal Convolution Network(TCN) model, namely Context-Aware Temporal Convolution Network(CATCN), which utilized local sub-segments to portrait the potential patterns of a series based on series decomposition. The experiments were conducted using three sets of field-captured traffic datasets. The results were presented and compared to state-of-the-art methodologies. The results showed that the performance of proposed method is significantly improved, especially, on the auto-correlation series corpora.

INDEX TERMS Traffic prediction, series decomposition, local context awareness, temporal convolution network.

I. INTRODUCTION

In recent years, highway traffic flow prediction has gained increasing attention, as monitoring the conditions of road networks is important in establishing intelligent transportation systems(ITS). It can be used to provide a considerable amount of information for road operators to evaluate the current traffic pattern so that traffic congestion or severe traffic accidents might be predicted in advance. However, traffic prediction is a challenging task due to the complex and dynamic characteristics of traffic.

To resolve this problem, statistical models, were widely considered by researchers at earlier stages, including auto regressive integrated moving average(ARIMA) [1] and support vector regression (SVR) [2], [3]. These methods were proposed under the conditions of insufficient computational power and data for analysis, and therefore, they encountered difficulties in capturing high-dimensional and nonlinear characteristics. Alternatively, researchers focused on deep learning models such as a long short-term memory neural network (LSTM) in traffic speed prediction [4], [5]. A hybrid comprising a fuzzy neural network (EFNN) and a Gaussian fuzzy membership function was introduced to predict the traffic speed [6]. A traffic graph convolution LSTM neural network(TGC-LSTM) was proposed to estimate traffic graph convolution based on a physical network topology combined with LSTM to improve the prediction performance [7]. Although all aforementioned methods were investigated for traffic prediction, several issues were still existed: 1) Some methods required neighbouring information to be incorporated in neural networks. While this procedure could enhance the prediction capability, it also deteriorated

The associate editor coordinating the review of this manuscript and approving it for publication was Hiu Yung Wong.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

model performance as it required evaluating spatio-temporal effects of connected parts. In highway speed prediction, for example, no considerable neighboring effect is exhibited, as highway networks are generally not as complicated as city road networks. 2)Basically, traffic speed prediction can be considered as a task to predict the speed time series with seasonal patterns that can be extracted from prior data series. But in reality, time series are usually much more complicated, which makes capturing different patterns to be a challenging task. Meanwhile, abilities of deep learning models to pick up seasonality and trends from given series are still insufficient.

When predicting a time series data, it is more common to forecast the latter value via the seasonality of sequential data. Therefore, the values of segments in sequence play a significant role. In the present paper, we propose a novel Context-Aware based Temporal Convolution Network named CATCN to solve the traffic prediction problem. It implies extracting prior periodic knowledge and combining it with original sequence. In our study, traffic flow with periodical changes indicates that it has autocorrelation feature, suggesting that its variation patterns can be easily grasped. Finally, in the conducted experiments, we find that the separated data comprising the information about the observed traffic flow with explicit periodic changes provide better capability. This confirms that the idea of including prior periodic knowledge is deemed. The key contributions of this research include the following:

- We propose a mechanism to extract the importance features of every point under its micro-context condition, considering both seasonality of a global series and its local neighbors. We choose several classical decomposition methods to compute the correlation between a sample and a target area in corpora.
- The proposed model leverages both micro-context sensitivity and global longer periodic dependencies. Unlike other spatio-temporal based approaches, our CATCN model does not need additional features and utilize only the generated decomposition features by a series itself.
- The result of evaluating the proposed method on three real-world traffic datasets demonstrate CATCN provides better capability of capturing patterns in a series. The proposed method achieves the lowest forecasting error compared with four state-of-the-art methods.

The rest of this paper is be organized as follows. Section II provides an overview on the related research works dedicated to time series forecasting and traffic flow prediction. In Section III, we first describe the overall architecture of the proposed framework and then introduce the detailed building modules that include the determination of a sliding window, context-aware feature generation, and context-aware convolution. In Section IV we discuss the results of the experiments conducted on three different datasets and compare the performance of the proposed method and the alternative approaches. Eventually, in Section V we conclude on the results acquired from experiments and summarize the overall contribution of this research.

II. RELATED WORK

A. TIME SERIES FORECASTING

As one of the most commonly used models in machine learning, time series forecasting could be applied in various fields [8], [9]. In recent years, due to the characteristics and the basic utilization of traffic flow prediction, it has been considered as a time series forecasting problem.

Earlier methods, such as ARIMA [10] or XGBoost [11], are widely used in time series tasks. Due to its mathematical soundness, ARIMA can achieve an acceptable performance [12]–[15] and can be combined with the other neural networks to further upgrade its performance [16], [17]. XGBoost is frequently used with the combination of other modules so that advantages of each modules can be integrated and yielding better results [18]–[20]. With the rapid evolution of deep learning frameworks, the time series forecasting problem is mostly considered from the viewpoint of neural networks, including LSTM [21], [22] and WaveNet [23] that has been initially designed for audio generation [24]. Among other methods, we can consider like TCN that is deemed applicable to specific issues, and simply relies on the dilation convolution capturing longer temporal information with a growing reception field. TCN ignores the local periodic characteristics of convolution features [25]. Other hybrid TCN approaches, e.g., Multi-Stage TCN (MS-TCN), Ensemble Empirical Mode Decomposition-Temporal Convolutional Network (EEMD-TCN) and Temporal Graph Convolutional Network (T-GCN) are integrating external information to help improving the forecasting capacity [26], [27].

B. EXISTING MODELS FOR TRAFFIC PREDICTION

A non-convex low-rank plus sparse decomposition model attempts to separate the rearranged matrix into low-rank and sparse matrices. Therefore, the resulting non-convex optimization problem can be efficiently handled using the augmented Lagrange multiplier (ALM) algorithm [28]. Meanwhile, several existing methods were investigated to apply convolutional neural networks(CNN) to traffic prediction owing to the recent advancement of the CNN-related networks and their excellent performance. Wu *et al.* proposed a model defined as a mixture of CNN and LSTM [29]. The model relied on the powerful feature extraction ability of CNN and considered the characteristic of the traffic prediction problem through LSTM. Fusion convolutional LSTM network(FCL-Net) was proposed to integrate the spatial and temporal dependencies [30]. A model called ITRCN attempted to convert a traffic network into images and apply a CNN to extract underlying characteristics. Moreover, it processed temporal features by using the gated recurrent unit(GRU) [31]. The methods based on CNN are deemed more capable of capturing spatial dependencies. However,

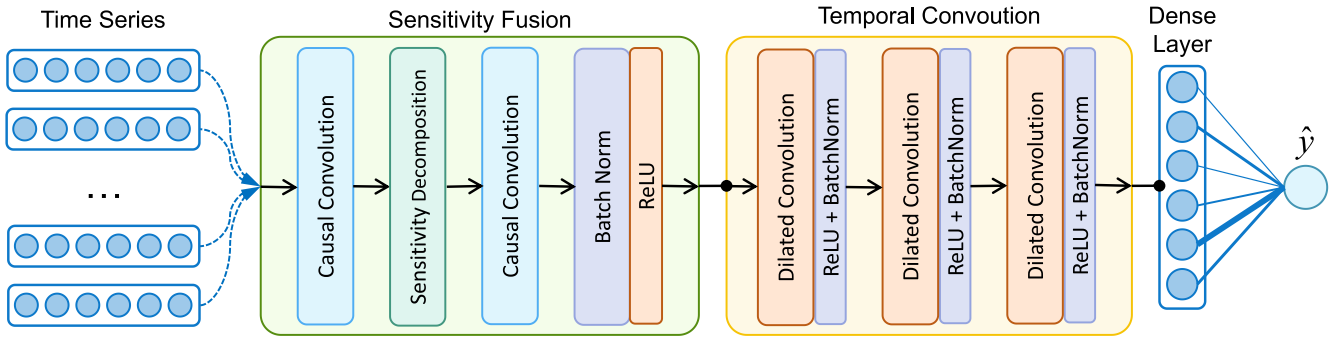


FIGURE 1. The architecture of CATCN Model.

CNN may fail to consider the locality of sub-segments simultaneously, local periodicity might be ignored.

A graph convolution network(GCN) can be used to simulate dependence between connecting neighbors in the non-Euclidean space and aggregate the spatial information of related nodes [32]. Although GCN allows incorporating the impact of a neighborhood into prediction, while the dependence on a pre-defined graph structure makes it unstable for dynamic scenario. Zhao *et al.* [33] proposed a model combining GCN and GRU called T-GCN that also captured both temporal and spatial relations. Many researchers have already found that the structure of a road and its connected nodes, such as interchange or toll stations, provides relevant information. Zheng *et al.* [34] proposed the generative multi-advanced network(GMAN) that utilized multiple attention blocks to model the impacts of spatio-temporal factors on prediction performance.

III. FRAMEWORK

To address the traffic prediction problem, we propose a novel framework that is based on fusing local structure contexts and global trends to enable the model to better capture series patterns. The overall architecture is illustrated in Figure 1. Sensitivity fusion captures periodic local dependencies and combines prior knowledge with the original series. Causal convolution can ensure the consistency of the channels during the fusion process. Sensitivity decomposition module decomposes the trend and seasonal components of the original series that can preserve the global features. Meanwhile, the method generates VSD, VMD and DPR vectors which preserve the local features. Then, these features are integrated by another causal convolution.Receptive fields grow exponentially as the layer deepens, enabling dilated convolution to extract both global and local patterns. After propagation through a dense layer, the network forecasts a value $\hat{y}$. It should be noted that the thickness of the last full connection varies, reflecting that the significance of each point is fused and has different weights.

Specifically, we apply casual convolution to transform a channel dimension and to establish interaction between channels. The segment levels of the original series are acquired through the sampling process. Therefore, by estimating local contexts, we can extract the context-aware vectors with the same length as the original series. The capability of capturing the longer periodic context awareness information in a neural network is ensured by utilizing stacked dilated convolution that has exponentially growing reception fields. The network outputs the forecasting results one point at a time after executing propagation through dense layers.

A. LOCAL CONTEXTS GENERATION

As the length of each context can vastly influence the capability of the proposed model to capture series patterns, we need to determine the length of each context first. We define a sliding window W that describes every single context's length throughout a sampling process. Given a series as following:

$$S = \{x_1, x_2, x_3, \ldots, x_n, x_{n+1}\} \tag{1}$$

in which n denotes the length of the series and x_{n+1} denotes its next timestamp. We first determine the upper limit of the size of sliding window $|W|$ using moving average.

Moving average plays the role of a low-pass filter that eliminates the high-frequency disturbance in a time series and maintains the useful low-frequency trend. Low-frequency filtering at time t turns to the convolution of time series S after adding a window with length $|W|$. Filtering function $\mathcal{F}$ in this window is defined as follows:

$$x_t = \sum_{i=t-(|W|-1)/2}^{t+(|W|-1)/2} \mathcal{F}_i y_{t-i} \tag{2}$$

in which $|W|$ denotes the size of a sliding window; $\mathcal{F}$ denotes the filtering function; x_t denotes a point at timestamp t.

For each point $x_i \in S$, we compute the moving average with window size varies from 1 to n. The upper limit of the sliding window size W_i^{up} at the i-th position is reached when the mean absolute percentage error reaches the minimum. Therefore, we obtain a series of upper limits of $|W|$:

$$W^{up} = \{W_1^{up}, W_2^{up}, \ldots, W_n^{up}\} \tag{3}$$

We perform grid search $|W|$ with upper limit W_i^{up} for each $x_i \in S$ by computing the autocorrelation coefficient as

Algorithm 1: Sensitivity Fusion

Input: Time series $S \leftarrow \{x_1, x_2, x_3, \ldots, x_n\} \in \mathbb{R}^{n_c \times n}$ with channel dimension n_c and length n.
Output: The series fuses local context S'.
Data: Sensitivity vector $V_{VSD} \leftarrow \varnothing$, $V_{VMD} \leftarrow \varnothing$, $V_{DPR} \leftarrow \varnothing$. Determine the sliding window size $|W|$ through the method mentioned in Section III.A.

STL time series decomposition $S_e, T_r \leftarrow STL(S)$;
Causal convolution with kernel size $K_1 \in \mathbb{R}^{1 \times n_c \times 1}$, $S \leftarrow conv(S, K_1)$;
Apply zero-padding to series S, and pad $\frac{|W|}{2}$ zeros to the head and the tail of the series respectively ;
Sample the tail of series S to obtain target context $S_t \leftarrow \{x_{n-W+1}, x_{n-W+2}, \ldots, x_n\}$;
Duplicate target context n times to gain target vector V_t ;
for *i=1:n* **do**
    Sample series S by sliding window to obtain local contexts ;
    $S_i \leftarrow \{x_{i-\frac{|W|}{2}}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+\frac{|W|}{2}}\}$;
    $S_{VSDi} \leftarrow \frac{(S_i - S_t)^2}{|W|}$;
    $S_{VMDi} \leftarrow \frac{|S_i - S_t|}{|W|}$;
    $S_{DPRi} \leftarrow \frac{(S_i \times S_t)}{S_t}$;
    Add S_{VSDi},S_{DPRi},S_{VMDi} to S_{VSD},S_{VMD},S_{DPR} repsectively ;
end
Extend series in channel dimension $S \leftarrow S \oplus S_{VSD} \oplus S_{VMD} \oplus S_{DPR} \oplus S_e \oplus T_r$;
Causal convolution with kernel size $K_2 \in \mathbb{R}^{n_c \times d_{c1} \times 1}$, $S' \leftarrow Conv(S, K_2)$;

follows:

$$\rho_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{4}$$

where $\bar{x}$ denotes the mean value; k represents the range of autocorrelation computation. The optimal $|W|$ at the i-th position can be determined using the following rule:

$$w_i = k \quad if \quad \rho_k = max\{\rho_j\}, j \in [1, W_i^{up}] \tag{5}$$

where w_i denotes the optimal sliding window size $|W|$ at the i-th position. Eventually, the final sliding window size $|W|$ can be determined by voting. Accordingly, the most votes may correspond to be the window size.

Let us suppose that x_{n+1} is the point to predict for given series S; then, sample context is defined as follows:

$$S_t = \{x_{n-|W|+1}, x_{n-|W|+2}, \ldots x_n\} \tag{6}$$

Then we sample each point to extract local context as follows:

$$S_i = \{x_{i-\frac{|w|}{2}}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+\frac{|w|}{2}}\}, i \in [1, n] \tag{7}$$

while sliding the window through the entire series for the point. In each sample context, we exclude the sample point, meaning that we only consider neighbor points as its context. It should be noted that if the length of a sample context is less than $|W|$, then zero-padding is applied to keep the length fixed. After sampling, we can obtain one target context and n sample contexts. In vectorization representation, the target context can be represented as a target vector $V_t \in \mathbb{R}^{1 \times |W|}$, and all sample contexts can be represented as a vector $V_s \in \mathbb{R}^{n \times |W|}$.

B. CONTEXT-AWARE VECTOR GENERATION

After context generation, we obtain $|W|$ features for each context, and then we apply the sub-series data corresponding to each time series are mapped to points in the $|W|$-length space. Therefore, the historical locality of a time series can be preserved in this way, including the dimension of the series and the complexity of computations.

For each sample context, we apply three different methods to compute the similarity between itself and the target context.

1) VALUE SQUARE DEVIATION (VSD)

$$VSD(S_i, T_t) = \frac{1}{|W|}\sum_{j=1}^{|W|}(S_{ij} - T_{tj})^2 \tag{8}$$

where $|W|$ denotes the sliding window size; S_{ij} is the j-th value of local context S_i; T_{tj} corresponds to the j-th value of target context T_t. VSD measures the average square deviation between two contexts.

2) VALUE MEAN DEVIATION (VMD)

$$VMD(S_i, T_t) = \frac{1}{|W|}\sum_{j=1}^{|W|}|S_{ij} - T_{tj}| \tag{9}$$

VMD measures the average mean deviation between two contexts.

3) DOT PRODUCT RATIO (DPR)

$$DPR(S_i, T_t) = \frac{\sum_{j=1}^{|W|} S_{ij} \times T_{tj}}{\sum_{j=1}^{|W|} S_{ij}^2} \tag{10}$$

DPR is used to measure the ratio of the dot product between two contexts, and the value range is $[\frac{n-2}{2n-1}, 1]$.

Then we decompose the series using STL, a filtering procedure for decomposing a time series into trend, seasonal and remaining components based on loess [35]. The STL decomposition comprises two recursive procedures: one is inner loop and the other is outer loop. In detail, the inner loop consists of six steps: detrending, cycle-subseries smoothing, low-passed filtering of smoothed cycle-subseries, detrending of smoothed cycle-subseries, deseasonalizing and trend smoothing. Therefore, the decomposed trend and seasonality of the time series are representative features that can reflect the overall characteristics of the series.

A time series can be regarded as superposition of different components $Y = T_r + S_e + R_e$, where Y denotes the original

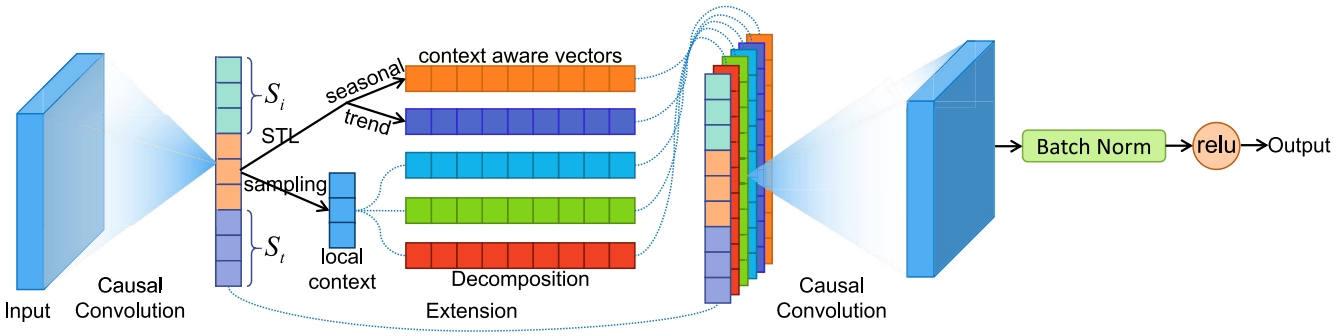


FIGURE 2. Structure of the sensitivity fusion layer. Causal convolution is used to perform channel transformation and interaction. After decomposition, we extend the original series using the sensitivity vectors in the channel direction for fusing. It should be noted that the dimension of the series before and after sensitivity fusion remains the same, only except the fact that the local periodic pattern is included.

time series; T_r refers to the trend component; S_e corresponds to the seasonal components; R_e is the remainder. A detrended series can be referred to as $Y - T_r$. Then, each cycle-subseries is smoothed by loess at all time positions. The collection of smoothed values for all cycle-subseries comprise the temporary seasonal series C. The low-pass filtering smoothing outputs L. Next, the seasonal component is subtracted by $S_e = C - L$. Then the deseasonalizing is applied by $Y - S_e$. As a result, components of the time series are extracted.

To fuse the context-aware vectors generated by the methods mentioned above, we concatenate them based on the original series in the channel direction:

$$S' = S \oplus S_{VSD} \oplus S_{VMD} \oplus S_{DPR} \oplus S_e \oplus T_r \tag{11}$$

where $\oplus$ denotes concatenation; S_{VSD} corresponds to the VSD series; S_{VMD} is the VMD series; S_{DPR} denotes the DPR series; S_e refers to the seasonal component of the series; T_r represents the trend component of the series. After channel extension, the original time series incorporates the local periodic information as its prior knowledge.

C. CONTEXT-AWARE CONVOLUTION

After the generation of context-aware vectors, we focus on temporal convolution to consider the local periodic information and accordingly to make more reasonable predictions. Context-aware convolution comprises three major steps.

Step 1 (Sensitivity Fusion): In the proposed model, we check whether it is required to compress channels and provide interactions between different channels by applying causal convolution before sensitivity fusion. This is because both the number and the length of channels corresponding to series would change once a convolution computation is applied. The aim is to realize end-to-end learning. As illustrated in Algorithm 1, applying causal convolution could ensure the consistency of channels, expanding the channel of a context-aware series that can be used for further training. The structure of the sensitivity fusion layer is presented in Figure 2.

Step 2 (Temporal Convolution): At this step, we enlarge the receptive field by stacking three dilated convolution layers with dilation equal to 1, 2, 4, ensuring that the longer periodic context awareness information is captured during the process. We aim to enable the network to predict based on different point weights, and therefore, we need to enhance the context importance through each layer. This can be achieved by fusing the sensitivity information with the current series before performing each dilated convolution. This step is illustrated in Algorithm 2.

Algorithm 2: Context-Aware Temporal Convolution

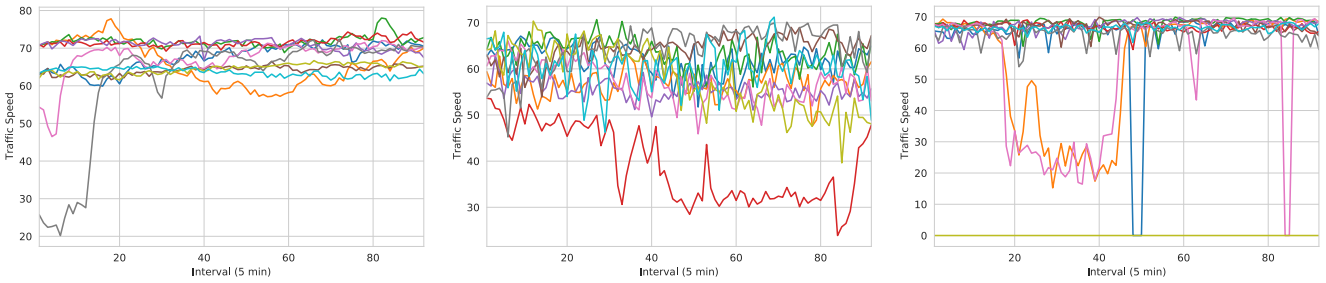
Input: Time series $S \leftarrow \{x_1, x_2, x_3, \ldots, x_n\} \in \mathbb{R}^{n_c \times n}$ with channel dimension n_c and length n
Output: Context-aware series $\mathbb{S}$
Data: Convolution layers n_l, dilation sizes $l \leftarrow \{1, 2, 4, \ldots, 2^{n_l-1}\}$, convolution kernels $K \leftarrow \{K_1 \in \mathbb{R}^{d_{c1} \times d_{c2} \times 1}, K_2 \in \mathbb{R}^{d_{c2} \times d_{c3} \times 1}, \ldots, K_{nl} \in \mathbb{R}^{d_{cnl} \times d_{cnl+1} \times 1}\}$

```
for i=1:n do
    Sensitivity fusion, S' ← SensitivityFusion(S) ;
    Dilated convolution computation with kernel K_i,
    𝕊 ← Conv(S_i, K_i, dilation = l_i) ;
    S ← 𝕊 ;
end
```

Step 3 (Forecasting): To update the value of the series, the learned features are propagated through the dense layer at the last step. The original series were fused with prior knowledge through temporal convolution so that the points in the dense layer have different weights, and therefore, the network can approach the forecasting result in accordance to the real values automatically.

D. MODEL TRAINING

As illustrated in Figure 1 and Algorithm 2, dilated convolution is one of the key components of the proposed model. The dilated convolution operator can apply the same filter with different ranges using various dilation factors. Let d be



(a) PEMS-BAY *(b) Seattle-ILDD* *(c) Metr-LA*

FIGURE 3. Ten random selection of a time series of three datasets.

a dilation factor and $*_d$ is defined as:

$$(F *_d k)(p) = \sum_{s+dt=p} F(s)k(t) \tag{12}$$

where $*_d$ represents a dilated convolution or an d-dilated convolution; F is a discrete function; k is a discrete filter; p refers to the receptive field.

Then the proposed model updates the weights to minimize the cost function by backpropagation. Suppose $\delta^{(l+1)}$ represents the error term for the $(l+1)$-st layer in the network with a cost function $J(W, b; x, y)$ where (W, b) are the parameters and (x, y) are the training data and ground-truth values. If the l-th layer is densely connected to the $(l+1)$-st layer, then the error for the l-th layer is computed as:

$$\delta^l = \left((W^{(l)})^T \delta^{l+1}\right) \bullet f'(z^{(l)}) \tag{13}$$

and the gradients are:

$$\nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (\mathbb{S}^{(l)})^T \tag{14}$$

$$\nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)} \tag{15}$$

Eventually, to calculate the gradient with respect to the filter maps, we rely on the border handling convolution operation again and flip the error matrix $\delta_k^{(l)}$:

$$\nabla_{W_k^{(l)}} J(W, b; x, y) = \sum_{i=1}^{m} (\mathbb{S}_i^{(l)}) * rot90(\delta_k^{(l+1)}, 2) \tag{16}$$

$$\nabla_{b_k^{(l)}} J(W, b; x, y) = \sum_{a,b} (\delta_k^{(l+1)}) \tag{17}$$

where $\mathbb{S}^{(l)}$ is the input to the l-th layer and the temporal convolution output of the $(l-1)$-th layer; $rot90$ denotes rotation of ninety degrees. The operation $(\mathbb{S}_i^{(l)}) * \delta_k^{(l+1)}$ is the "valid" convolution between i-th input in the l-th layer and the error with respect to the k-th filter.

IV. EXPERIMENTS

In this section, we mainly describe the setup of the conducted experiments and compare the performance of the proposed CATCN with several existing deep learning models that serve as baselines in traffic flow prediction.

A. DATASET DESCRIPTIONS

In the experiments, we use the following datasets to test the performance of the proposed model. To explicitly reveal the peculiarity of a traffic time series, we randomly extracted ten examples from all three datasets. First, we compute basic properties of each series, such as auto-correlation, mean change, mean second derivative central etc. Then, we cluster the series into five clusters, from each cluster we randomly select two examples and put all the examples together for plotting. In this way, we can make sure that the randomly selected subset of the data is representative enough.

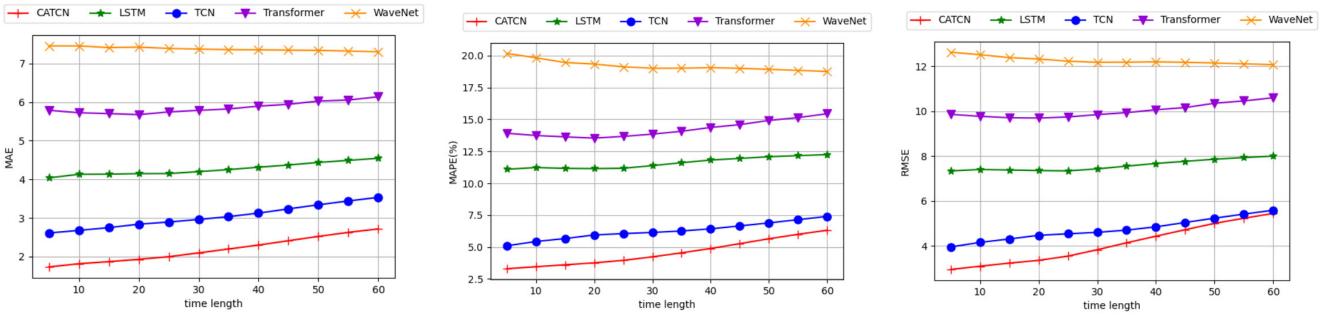
1) PEMS-BAY*

This dataset was obtained from the California department of transportation. It contains the description of the road occupy rate corresponding to the Los Angeles County highway network. The dataset comprised the information about the traffic speed registered by 325 sensors in the Bay Area of California, starting from January 1 2017, to May 31 2017. We aggregated the observed traffic speed values into five-minute intervals having the size of 6030×92. Then we separate the dataset into observation group 6030×80 for training and forecasting group 6030×12 for forecasting. Std of the dataset is 8.72; mean value 62.94; min value 3.70, max value 76.90. As shown in Figure 3 (a), most of the recorded patterns varied periodically in time, while some of them demonstrated abrupt jumps in the beginning. Theoretically, a time series of a traffic flow with recurrent changes should have represented more accurate results.

2) SEATTLE-ILDD†

The data was collected by using inductive loop detectors deployed on freeways in Seattle area. The freeways contained I-5, I-405, I-90, and SR-520. This dataset contained the spatio-temporal information about the speed of the considered freeway system. The speed information at a milepost was averaged over the data from multiple loop detectors on the main lanes in a same direction. The dataset is aggregated into five-minute intervals with the dimension 5730×92. It is

*https://github.com/liyaguang/DCRNN
†https://github.com/zhiyongc/Seattle-Loop-Data



(a) MAE (b) MAPE (c) RMSE

FIGURE 4. Results of the performance evaluation on the PEMS-BAY dataset in terms of three metrics. The horizontal line represents the time length and the vertical line indicates one of the tested metrics.

TABLE 1. Context-aware TCN parameter settings.

Learning rate λ	0.001	Weight Decay	0.0001	Batch size	256
Dilation factors	[1,2,4]	Sliding window size W	4	Extend dimension	4
Dense layers	2	Units in dense layers	[73,12]	Kernal size	2

separated into observation group 5730 × 80 for training and forecasting group 5730 × 12 for forecasting. Std of the dataset is 8.14; mean value 59.94; min value 3.59, max value 75.94. The random sampling results are represented in Figure 3 (b).

3) Metr-LA[‡]

Metr-LA was a dataset comprising the information from the Los Angeles highway. Specifically, the dataset contained the data on the traffic speed registered during four months using 207 sensors deployed in the county. The dimension of this dataset is 1000 × 92 with five-minutes interval. It is separated into observation group 1000 × 80 for training and forecasting group 5730 × 12 for forecasting. Std of the dataset is 19.19; mean value 58.55; min value 0.00, max value 70.00. According to the random sampling results represented in Figure 3(c), both sudden speed changes and static time series could be observed.

For each dataset, we run and evaluate all the methods ten times to eliminate outliers and then average the results to reduce random error. We apply Z-score normalization and split the dataset into a training set (70%) and test set (30%) in a chronological order randomly during each run, enabling the experiments to be conducted in a rigorous and controlled environment to make it generalizable.

B. COMPARISON WITH THE BASELINE METHODS

To prove the validity of the proposed approach, we compared four forecasting methods: 1) neural network-based methods, including LSTM, Transformer, TCN, and WaveNet; 2) neural networks integrated with context awareness: CATCN (the proposed method).

[‡]https://www.metro.net/

C. EVALUATION METRICS

To compare the performance and the effectiveness of the considered methods, we utilized the following metrics: mean absolute error(MAE), mean absolute percentage error(MAPE) and root mean square error(RMSE).

MAPE is used to measure the relative errors, and is often reported as a percentage:

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \tag{18}$$

where y_i denotes the prediction output; $\hat{y}_i$ is the ground-truth value; n corresponds to the total length of the series.

MAE is applied to measure the average absolute error between the predicted value and the ground-truth value and is calculated as follows:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{19}$$

RMSE was employed to measure the deviation between the predicted and ground-truth values. RMSE was selected as it deemed more sensitive to outliers:

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{20}$$

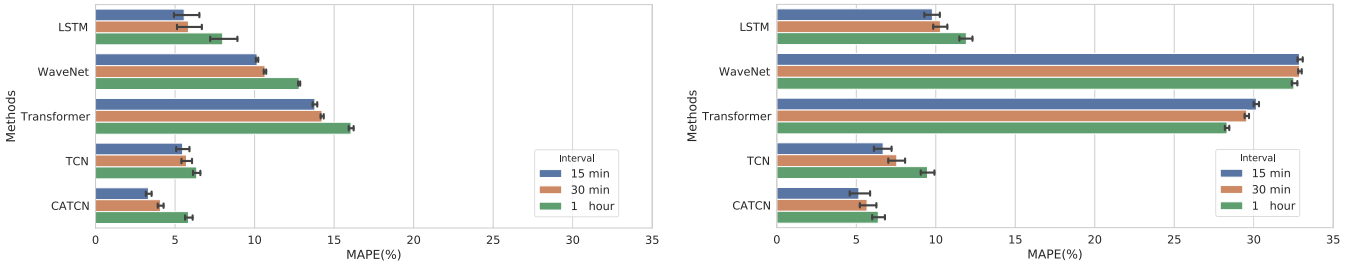
In above equations, y denotes the ground-truth value and $\hat{y}$ denotes the predicted value outputted by the network.

D. PARAMETER SETTINGS

In the considered benchmark models, we used the following parameter settings:

LSTM [36]: hidden dimension $d^h = 10$ with one layer stacked;

WaveNet [24]: residual channel 32; skip channel 128 with layer $K = 4$ for each block; three blocks stacked in total;



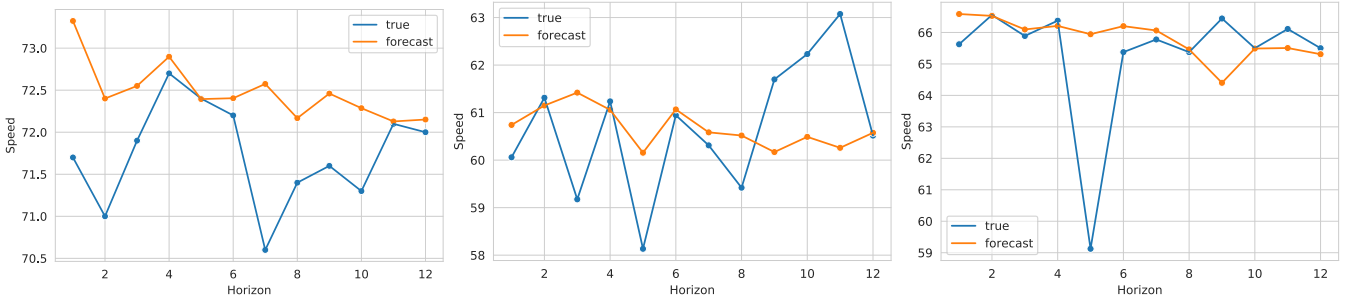
(a) Variance of the subset of PEMS-BAY with high autocorrelation

(b) Variance of the subset of PEMS-BAY with low autocorrelation

FIGURE 5. **Variance estimation of five methods on PEMS-BAY when autocorrelation values of the time series are different:(a) shows the results when autocorrelation is in [0,0.6] and (b) represents the rest of cases. Each bar in the figure refers to the mean MAPE averaged on ten runs. Meanwhile, the standard deviation of each forecasting interval is presented.**

TABLE 2. **Performance evaluation on 15 mins ahead prediction on three datasets.**

Dataset	PEMS-BAY			Seattle-ILDD			Metr-LA		
Methods	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE
LSTM(2015)	11.17%	4.13	7.38	5.44%	2.74	4.29	9.95%	3.89	7.39
WaveNet(2016)	19.46%	7.41	12.39	10.30%	5.10	7.13	11.61%	4.73	8.34
Transformer(2017)	13.63%	5.70	9.71	10.50%	5.24	7.29	12.31%	4.60	8.50
TCN(2019)	5.67%	2.75	4.30	6.50%	3.11	4.69	11.40%	4.33	7.10
CATCN	**3.61%**	**1.87**	**3.23**	**4.41%**	**2.11**	**3.54**	**7.75%**	**3.29**	**6.59**



(a) PEMS-BAY

(b) Seattle-ILDD

(c) Metr-LA

FIGURE 6. **Visualization of forecasting result of CATCN on three datasets.**

Transformer [37]: eight layers in total; query size $q = 32$; value size $v = 32$; the number of heads 32; hidden dimension $d^h = 256$; attention window size 32; dropout rate $\beta = 0.3$. The rest of parameter settings remained the same as in the original paper;

TCN [38]: three dilated convolution layers stacked; each layer had the kernel size 2 and stride 1.

Parameters for the context awareness integrated model are provided in Table 1.

E. LOSS FUNCTION

To train the models through back propagation and to measure the deviation between the prediction and the ground-truth values, we adapted RMSE as the loss function:

$$loss(y, \hat{y}) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}} \tag{21}$$

where y_i denotes the prediction output; $\hat{y}_i$ is the ground-truth value; n corresponds to the total length of the series in question.

F. FORECASTING PERFORMANCE EVALUATION

Table 2 provides the forecasting results averaged on ten runs on three traffic datasets, **the best results** are highlighted in bold and the second-best results are underlined.

The accuracy of all tested methods applied to the PEMS-BAY dataset with the varying time length is represented in Figure 4. We illustrated the performance of the proposed method and the other four alternative approaches while extending the time length. As observed, except WaveNet, the tested models exhibited increasing errors as the time length augmented, and yet CATCN still outperformed other compared methods in terms of three metrics.

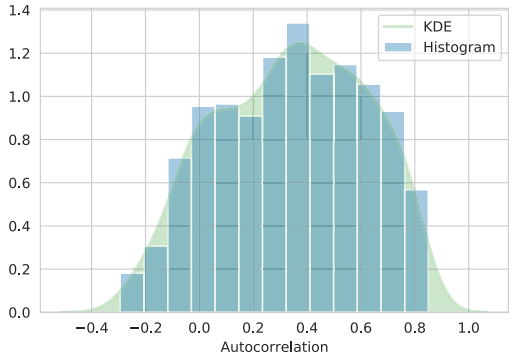


FIGURE 7. The autocorrelation of the PEMS-BAY dataset.

We demonstrated the relevance of autocorrelation and represent model performance, as shown in Figure 5 and Figure 7. Low autocorrelation of the considered time series indicated that the sequence did not reflect typical periodical changes and tended to vary without exhibiting explicit recognizable patterns, thereby hindering sequence prediction to achieve satisfying results. We can observe that when autocorrelation is in [0,0.6], MAPE on PEMS-BAY tends to have smaller variance and the outliers differ in short intervals.

Figure 6 provides the visualization of the forecasting results on the three datasets. We can notice that after integrating the context-awareness features, the proposed method has a better capability of capturing the local patterns. Meanwhile, even though the trend of the forecasting result is consistent with the true value in general, the proposed CATCN still meets challenges when the abrupt change occurs.

V. CONCLUSION

In the present paper, we proposed a novel deep learning architecture that was capable of performing local features extraction and combining prior knowledge with the original series to achieve better performance in traffic prediction compared with the existing methods. It should be noted that the proposed network relied on a generic method and therefore, it could not only achieve better results being applied to in traffic series but also was expected to perform well in general time series forecasting. The proposed CATCN method could learn a pattern of the local fluctuation and enhance performance after extending the channels of time series with periodic trends. The end-to-end training was realized by integrating causal and dilated convolution, thereby improving the robustness of the proposed network.

The results of the conducted experiments indicated that the proposed CATCN achieved the best results compared with the considered alternative methods and also demonstrated that integrating traffic time series with local sensitivities allowed capturing useful information. Furthermore, the proposed method did not require to train attention weights, still providing better capabilities compared with the method using attention mechanisms.

REFERENCES

[1] H. Chen and S. Grant-Muller, "Use of sequential learning for short-term traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 9, no. 5, pp. 319–336, Oct. 2001.
[2] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transp. Res. Rev.*, vol. 7, no. 3, p. 21, Sep. 2015.
[3] P. V. V. K. Theja and L. Vanajakshi, "Short term prediction of traffic parameters using support vector machines technique," in *Proc. 3rd Int. Conf. Emerg. Trends Eng. Technol.*, Nov. 2010, pp. 70–75.
[4] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transp. Res. C, Emerg. Technol.*, vol. 54, pp. 187–197, May 2015.
[5] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*. [Online]. Available: http://arxiv.org/abs/1801.02143
[6] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Trans. Intell. Transport. Syst.*, vol. 18, no. 9, pp. 2340–2350, Sep. 2017.
[7] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Trans. Intell. Transport. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.
[8] Y. Li, R. W. Liu, Z. Liu, and J. Liu, "Similarity grouping-guided neural network modeling for maritime time series prediction," *IEEE Access*, vol. 7, pp. 72647–72659, 2019.
[9] Y. Li, R. W. Liu, Q. Ma, and J. Liu, "Emd-based recurrent neural network with adaptive regrouping for port cargo throughput prediction," in *Proc. ICONIP*. Cham, Switzerland: Springer, 2018, pp. 499–510.
[10] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003.
[11] C. Li, Z. Chen, J. Liu, D. Li, X. Gao, F. Di, L. Li, and X. Ji, "Power load forecasting based on the combined model of LSTM and XGBoost," in *Proc. Int. Conf. Pattern Recognit. Artif. Intell. (PRAI)*, 2019, pp. 46–51.
[12] M. H. Amini, A. Kargarian, and O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," *Electr. Power Syst. Res.*, vol. 140, pp. 378–390, Nov. 2016.
[13] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Proc. 16th Int. Conf. Comput. Modeling Simulation (UKSim-AMSS)*, Mar. 2014, pp. 106–112.
[14] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud Applications' QoS," *IEEE Trans. Cloud Comput.*, vol. 3, no. 4, pp. 449–458, Oct. 2015.
[15] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "ARIMA models to predict next-day electricity prices," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1014–1020, Aug. 2003.
[16] D. Ömer Faruk, "A hybrid neural network and ARIMA model for water quality time series prediction," *Eng. Appl. Artif. Intell.*, vol. 23, no. 4, pp. 586–594, Jun. 2010.
[17] H. Liu, H.-Q. Tian, and Y.-F. Li, "Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction," *Appl. Energy*, vol. 98, pp. 415–424, Oct. 2012.
[18] J. Nobre and R. F. Neves, "Combining principal component analysis, discrete wavelet transform and XGBoost to trade in the financial markets," *Expert Syst. Appl.*, vol. 125, pp. 181–194, Jul. 2019.
[19] Y. Wang and Y. Guo, "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost," *China Commun.*, vol. 17, no. 3, pp. 205–221, Mar. 2020.
[20] W. Yucong and W. Bo, "Research on EA-xgboost hybrid model for building energy prediction," *J. Physics: Conf. Ser.*, vol. 1518, Apr. 2020, Art. no. 012082.
[21] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.
[22] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, Jan. 2019.
[23] D. Duan, "Research on hotel online sales forecast model based on improved WaveNet," *J. Phys., Conf. Ser.*, vol. 1544, May 2020, Art. no. 012067.

[24] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *Proc. ISCASSW*, 2016, pp. 1–15.
[25] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal convolutional networks applied to energy-related time series forecasting," *Appl. Sci.*, vol. 10, no. 7, p. 2322, Mar. 2020.
[26] Y. A. Farha and J. Gall, "MS-TCN: Multi-stage temporal convolutional network for action segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3575–3584.
[27] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI*, 2018, pp. 1–10.
[28] R. W. Liu, J. Chen, Z. Liu, Y. Li, Y. Liu, and J. Liu, "Vessel traffic flow separation-prediction using low-rank and sparse decomposition," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–6.
[29] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016, *arXiv:1612.01022*. [Online]. Available: http://arxiv.org/abs/1612.01022
[30] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.
[31] X. Cao, Y. Zhong, Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Interactive temporal recurrent convolution network for traffic prediction in data centers," *IEEE Access*, vol. 6, pp. 5276–5289, 2018.
[32] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proc. NIPS*, 2016, pp. 1993–2001.
[33] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transport. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
[34] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proc. AAAI*, 2020, vol. 34, no. 1, pp. 1234–1241.
[35] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, "Robuststl: A robust seasonal-trend decomposition algorithm for long time series," in *Proc. AAAI*, vol. 33, 2019, pp. 5409–5416.
[36] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. ESANN*, vol. 89, 2015, pp. 89–94.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, 2017, pp. 5998–6008.
[38] Y. Liu, H. Dong, X. Wang, and S. Han, "Time series prediction based on temporal convolutional network," in *Proc. IEEE/ACIS 18th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2019, pp. 300–305.

TAO RUAN received the Ph.D. degree in civil engineering from Clemson University, USA, in 2016. He is currently a Research Scientist with Zhejiang Institute of Transportation Company Ltd., Hangzhou, China. His research interests include artificial intelligence and data mining techniques for road traffic evaluation and safety analysis. He is a member of ASCE and NACE.

DEXING WU received the Ph.D. degree in bridge and tunnel engineering from Southwest Jiaotong University, in 2009. He is currently the CEO of Zhejiang Institute of Transportation Company Ltd. His research interests include innovative engineering design, road network optimization, and traffic data analysis.

TIANYI CHEN is currently pursuing the bachelor's degree in computer science and technology with the Zhejiang University City College. His research interests include implementation of machine learning algorithms and artificial intelligence.

CANGHONG JIN received the B.S. degree in software engineering and the M.S. and Ph.D. degrees in computer science from Zhejiang University, China, in 2005, 2008, and 2015, respectively. Since 2016, he has been an Assistant Professor of computer science with the Zhejiang University City College. He had extensive experience in big data system development, artificial intelligence applications, and software engineering management. His research interests include data mining and social network analysis, especially spatio-temporal data mining.

LEI XU received the master's degree in computer science from Zhejiang University, Hangzhou, China, in 2015. His research interests include system architecture, cloud computing, and so on.

SHENGLI ZHOU was born in Wenzhou, Zhejiang, China, in 1982. He received the Ph.D. degree from Army Engineering University, in 2018. He is currently an Engineer with the Zhejiang Police College. His research interest includes concern network security.

ZHEN JIANG received the B.S. degree in intelligence science and technology from Central South University. He is currently pursuing the master's degree with Zhejiang University. His research interests include machine learning, deep graph learning, and so on.