Acting as a Decision Maker: Traffic-Condition-Aware Ensemble Learning for Traffic Flow Prediction

Yuanyuan Chen, *Member, IEEE*, Hongyu Chen, Peijun Ye, *Member, IEEE*, Yisheng Lv, *Senior Member, IEEE*, and Fei-Yue Wang, *Fellow, IEEE*

***Abstract*— Accurate traffic prediction under various conditions is an important but challenging task. Due to the complicated non-stationary temporal dynamics in traffic flow time series and spatial dependencies on roadway networks, there is no particular method that is clearly superior to all others. Here, we focus on investigating ensemble learning that benefits from multiple base models, and propose a traffic-condition-aware ensemble approach that acts as a decision maker by stacking multiple predictions based on dynamic traffic conditions. To sense traffic conditions, we apply the Convolutional Neural Network (CNN) model to capture the spatiotemporal patterns embedded in traffic flow. Then, the high-level features extracted by CNN are used to generate weights to ensemble multiple predictions of different models. Extensive experiments are performed with a real traffic dataset from the Caltrans Performance Measurement System. We compare the proposed approach with competitive models, including Gradient Boosting Regression Tree (GBRT) model, Weight Regression model, Support Vector Regression (SVR) model, Long Short-term Memory (LSTM) model, Historical Average (HA) model and CNN model. Experimental results demonstrate that our method can effectively improve the performances of traffic flow prediction.**

***Index Terms*— Traffic flow prediction, ensemble learning, deep learning.**

I. Introduction

TRAFFIC prediction is an elemental function of Intelligent Transportation Systems (ITS) [1], as accurate and timely prediction is significant for both proactive traffic control and traveler information service. In recent years, traffic flow prediction has become one of the most popular topics in the field of ITS, and a lot of efforts have been made to enhance prediction accuracy by researchers and practitioners in ITS community.

Existing traffic prediction approaches can be roughly categorized into two types, i.e., model-based approaches and data-driven approaches. A model-based approach mimics the traffic dynamics by explicitly adopting traffic flow models (e.g., macroscopic model, mesoscopic model, or microscopic model). Once the traffic simulation system is well established, traffic prediction can be achieved through the evolution process of the system. In comparison, a data-driven approach establishes a mapping between historical traffic flow and its future prediction but does not presume a traffic flow model to describe the dynamics of a transportation system. With the development of big data and deep learning, data-driven prediction models have attracted more attention and achieved superior performances. However, as far as we can tell, there is no one traffic prediction model that consistently outperforms other models under any traffic condition and for any traffic dataset. The corresponding reasons may be two-fold: (1) non-stationary temporal dynamics in traffic flow time series and spatial dependencies on roadway networks yield complicated spatiotemporal characteristics that are hard to model, (2) a single model commonly has bias which makes it hard to be generalized under different conditions. Therefore, to address the generalization ability issue, ensemble models that can benefit from different base models were proposed for the challenging task of traffic prediction [2], [3].

Ensemble models have been proven to be effective in traffic prediction [4]. An ensemble model can be formulated as

$$F(x) = \sum_{m=1}^{M} \gamma_m f_m(x) \tag{1}$$

where F is an ensemble model, f are base models and γ are weights of base models. Most of previous studies apply fixed weights calculated in the stage of training regardless of the current traffic conditions. Taking the bagging approach as an example, the outputs of based models are directly averaged in traffic prediction task. However, the performances of different base models vary with different traffic conditions as illustrated in Fig. 1. From the three cases shown, it is obvious that fixed weights cannot achieve the best predictions. Therefore, the ensemble models with fixed weights limits their performances under different traffic conditions.

Manuscript received November 10, 2019; revised May 24, 2020 and September 27, 2020; accepted October 6, 2020. Date of publication November 10, 2020; date of current version March 29, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61903363 and Grant U1811463, and in part by the Early Career Development Award of State Key Laboratory for Management and Control of Complex Systems under Grant 20190205 and Grant 20190207. The Associate Editor for this article was L. Li. *(Corresponding author: Yisheng Lv.)*

Yuanyuan Chen and Yisheng Lv are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: yuanyuan.chen@ia.ac.cn; yisheng.lv@ia.ac.cn).

Hongyu Chen is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Automation, Harbin University of the Science and Technology, Harbin 150080, China (e-mail: chenhongyu2017@ia.ac.cn).

Peijun Ye is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Qingdao Academy of Intelligent Industries, Qingdao 266109, China (e-mail: peijun.ye@ia.ac.cn).

Fei-Yue Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Institute of Engineering, Macau University of Science and Technology, Taipa, Macau (e-mail: feiyue@ieee.org).

Digital Object Identifier 10.1109/TITS.2020.3032758

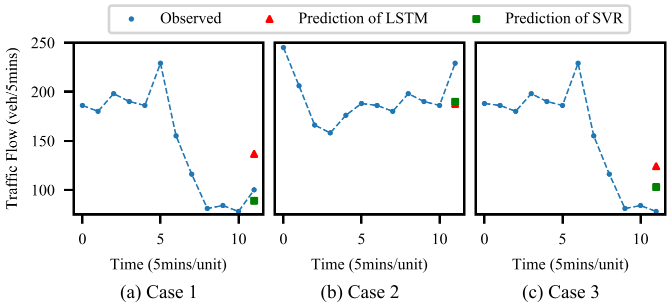


Fig. 1. Illustration of different performances of SVR and LSTM. (a) The prediction of LSTM overshoots its target, while that of SVR undershoots its target under case 1. (b) The predictions of LSTM and SVR both undershoots its target under case 2. (c) The predictions of LSTM and SVR both overshoots its target under case 3.

This paper proposes an ensemble learning framework, in which, to achieve better predictions under different conditions, the weights of base models can be dynamically adapted. In other words, our proposed approach attempts to make a decision on how to stack the outputs of base models according to their performances under certain traffic conditions. The motivation behind is that we assume that a higher-level model, focusing on making decisions to benefit from base predictor, can achieve better performances. As any existing model can be used as a candidate base model, our objective is to design the rules to generate the weights based on their historical performances. Therefore, we separate weights generation from base models training, and we focus on addressing the issue of weights generation in this paper.

In the proposed framework, traffic condition recognition and dynamic weights generation are the two essential modules. To sense traffic conditions, we develop a spatiotemporal representing learning model using Convolutional Neural Networks (CNN). The convolution operation is designed to extract temporal features by moving a filter, also as a feature detector, along the time dimension. After convolution operation, pooling operation is applied to extract spatial features by keeping the maximum value of different detectors that are spatially deployed. More details of convolution operation and pooling operation can be found in Section III. After one pair or multiple pairs of convolution operation and pooling operation, the features conveying spatiotemporal characteristics of traffic flow are further used to generate weights as a decision. We design a fully connected neural network to generate weights, which is used to combine the predictions of base models to make the final prediction.

The key idea of our approach is to use a model that can detect traffic patterns and evaluate the performances of base predictors to formulate weight coefficients that are used to ensemble the predictions of based predictors. The main contributions of this paper are summarized as follows:

1. We design an ensemble learning framework for traffic prediction, called Traffic-Condition-Aware Ensemble (TCAE), which incorporates the idea of making decisions on how to stack the predictions of base models with the information of traffic condition. A novel ensemble rule that uses traffic information to adaptively set the weights to combine based models based on their performances under different traffic conditions is proposed to further improve the performances of traffic prediction.
2. We formulate a CNN model to detect the dynamic traffic conditions by extracting spatiotemporal features embedded in traffic flow. A near-term traffic flow matrix is defined, and convolution operation on this matrix is designed to extract temporal features, then pooling operation is designed to extract spatial features.
3. We demonstrate experimentally that TCAE achieves better performances than both single model methods (Support Vector Regression Model, Long Short-Term Memory Model, and Historical Average Model) and ensemble methods (Weight Regression Model and Gradient Boosting Regression Tree Model).

The rest of this paper is organized as follows. In section II, we review recent advances in traffic prediction and especially ensemble approaches. In section III, we present our approach to ensemble predictions of base models by using CNN to extract spatiotemporal features. In section IV, we describe the experiments on a real traffic dataset to verify the effectiveness of our approach, and analyze the results. In section V, we conclude this paper and outlook potential research.

II. Literature Review of Short-Term Traffic Prediction Methods

Traffic prediction plays an important role in the field of ITS as mentioned above, and there are fairly extensive studies on models and algorithms for this problem [5], [6]. Generally, traffic prediction methods can be divided into two major categories, i.e., model-based approaches and data-driven approaches [7]. In this paper, we only focus on reviewing the data-driven approaches.

Data-driven traffic prediction approaches are more common in practice compared with model-based methods. The task of data-driven traffic prediction is to estimate unobserved traffic flow data by using observed data, including traffic flow data and exogenous traffic relevant data. And training a prediction model is to build a mapping of observed data to unobserved data.

A. Parametric and Non-Parametric Regression Approaches

Parametric regression models are earlier approaches used to predict traffic flow. Autoregressive Integrated Moving Average (ARIMA), proposed by statisticians George Box and Gwilym Jenkins, was used to predict traffic volume in 1980's [8]. ARIMA model has 3 components to model different types of patterns. It is symbolized as ARIMA(p,d,q) where "p" indicates the number of autoregressive terms, "d" indicates the order of differencing, and "q" indicates the number of moving average terms. And the first step is to identify these three model parameters by analyzing traffic flow series. Then estimation of coefficient parameters is conducted by minimizing the error term based on historical data. Subsequent researches have proposed its variants, such as Kohonen ARIMA, subset ARIMA, ARIMA with explanatory variables, and Seasonal ARIMA. Compared with ARIMA methods, Kalman filtering method, as another parametric regression approach, requires only limited input data. Generally, parametric regression methods are easy to apply, but not proper to be utilized to predict traffic flow series that vary quickly [9]. And non-parametric methods have advantages to handle the stochastic and non-linear characteristics of traffic flow, such as support vector regression (SVR) and k-Nearest neighbors (KNN). In SVR method, kernel function is used to model non-linear features by mapping historical traffic flow into feature space. Then the linear combination of features is outputted as the prediction of traffic flow. [10], [11]. KNN is another non-parametric approach in which the distance between training instances and the target instance is firstly calculated based on a distance function [12]. Then the closet k training instances are selected, and the average of their future values is outputted as the prediction value [13], [14].

B. Deep Learning Approaches

With the development of deep learning and big data, there have been considerable interests in using deep neural networks (DNN) to predict traffic conditions [15]. It is an end-to-end learning approach, which optimizes the network parameters by considering the inputs and outputs directly. Thus, it does not heavily depend on manual feature engineering, however, it remains a black box due to the lack of ability to interpret its behaviors. In practice, DNN approaches have generally achieved better performances than traditional parametric and non-parametric regression methods. The mainly used architectures are deep forward network (DFN), recurrent neural network (RNN) and convolutional neural network (CNN). Lv *et al.* proposed to use stacked auto-encoder (SAE) model to predict short-term traffic flow [16]. It is a DFN model and is trained in an unsupervised, greedy, layer-wise fashion. Unlike standard DFN, RNN has feedback connections between nodes form a directed graph along a temporal sequence [17], which allows it to model temporal dynamic behaviors of traffic flow series. Although it has potential to learn the temporal dependency, standard RNN is rarely used in practice due to the vanishing gradient problem [18]. Long Short-term Memory (LSTM) model and Gated Recurrent Unit (GRU), as variants of standard RNN, have achieved competitive performances. To speed up the training process and increase interpretability, a fuzzy neural network approach, which combines Type-2 Fuzzy Sets and LSTM, was proposed for traffic flow prediction [19]. There is a rising concern about the travel privacy when mining big data for mobility service. To address this issue, a Federated Learning-based GRU approach for traffic flow prediction was proposed [20]. Compared with RNN, CNN also have the ability to model spatial dependency between the data from different locations. CNN usually consists of convolutional layers interspersed with pooling layers. The convolutional layer aims to learn the region features by moving a filter over the input map and computing the dot products. And pooling operation computes the average or maximum of each region of the feature map to reduce following computational time and pass more abstract information to the next layers. Ma *et al.* proposed to use CNN to extract inter-day and intra-day traffic flow patterns [21]. In this study, traffic flow data is formulated as a matrix and regarded as an image with one channel. A similar study is [22], in which the road segments are regarded as nodes, and their topological relationships are considered to transform the road network into a compact 2D image. To extract the spatial features and semantic correlations of detectors, Lv *et al.* proposed to use a multi-graph convolutional network for traffic flow prediction [23] in which the convolution operation was conducted on a graph. An adversarial learning framework that combined CNN and LSTM was proposed for network-scale travel speed prediction [24].

C. Ensemble Learning Approaches

The basic idea of ensemble methods (e.g., bagging, boosting and stacking) is that firstly, a number of base models are trained in a parallel or sequential style. Then, a rule to combine the outputs of base models is developed to produce the final output [25]–[27]. In short, the training of base models and the constructing of combination rules are the core techniques of ensemble method. In the bagging approach, base models are trained based on different sets of training instances, then their outputs are directly averaged for regression or majority voting for classification [28]. In comparison, the boosting approach trains a series of base models and calculate their weights at the same time [29]. As for the stacking approach, a number of base models are first trained. Then a higher-level model or meta-model, which takes the outputs of base models as input, is trained to combine based models [30]. Generally, the bagging and boosting approach can also be categorized into the stacking approach according to the above description although the stacking approaches often ensemble heterogeneous base models in practical.

Ensemble methods cannot only reduce the variance of base models, but also enhance prediction accuracy [31]. In the field of traffic prediction, ensemble learning has achieved great success. An early study is that Chen *et al.* applied the bagging of radial basis function neural network to decrease the prediction error [32]. In recent studies, the gradient boosting regression tree (GBRT) model is frequently applied and achieves high performance. Chen *et al.* proposed an integrated framework of GBRT and least absolute shrinkage and selection operator (Lasso) for the short-term traffic flow prediction [3]. Similarly, Zhang *et al.* employed a GBRT model to predict freeway travel time [33]. Ma *et al.* used GBRT model to predict the incident clearance time based on different types of explanatory variables [4]. Yang *et al.* studied the influence of neighboring traffic condition on prediction performance, and applied GBRT model to make short-term traffic volume prediction [34]. Xiao *et al.* proposed an incremental regression framework under drifting environment, and utilized ensemble learning to update the distribution representation [35]. Ding *et al.* took bus transfer activities and temporal features into consideration, and applied gradient boosting decision tree to predict short-term subway ridership [36]. Zhang *et al.* extracted spatiotemporal correlations from historical and real-time traffic data for adjacent and target links, and utilized GBRT model to predict urban link travel time based on real-time and historical GPS records collected by probe vehicles [37]. Regarding the ensemble rule, a recent study proposed a weight integration strategy using population extremal optimization-based no negative constraint

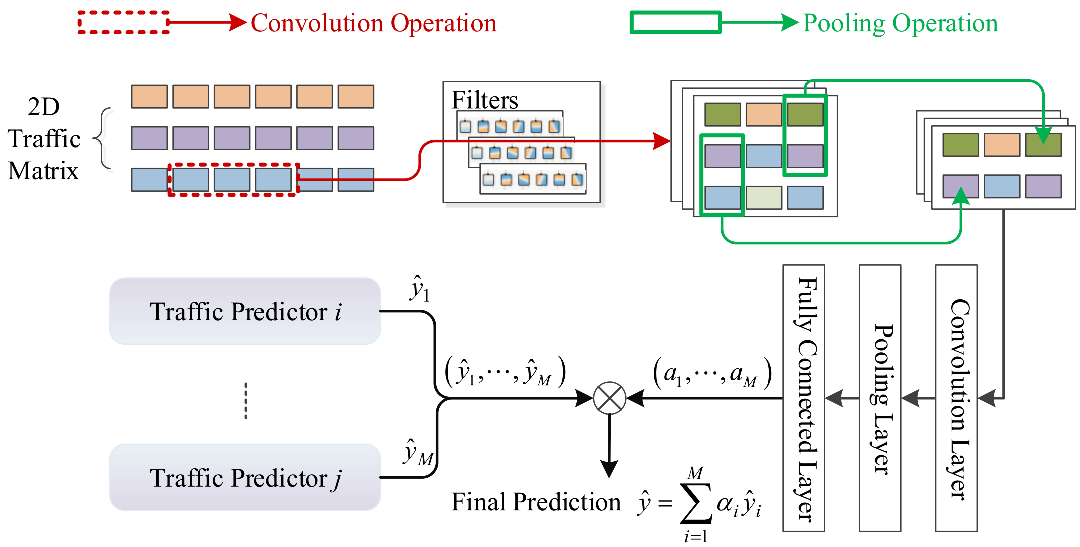


Fig. 2. Architecture of the proposed approach.

theory [38], in which only LSTM models are combined. Bayesian combination method was also proposed to calculate the weight coefficients [39], and this method also used fixed weights and failed to tackle the error magnification phenomena, i.e., the ensemble prediction error is larger than that of base models when the predictions of base models are all larger or smaller than the observed ground-truth value. Another study proposed an improved Bayesian combination method that use an error calibration coefficient to reduce the error magnification phenomena [40].

In summary, extensive researches on traffic prediction, especially short-term traffic flow prediction, have achieved considerable success. However, there is no one method that has been proved achieving superior performances than all other approaches under various conditions. Each base model has its advantages, and ensemble method by benefiting from base models can achieve better performance than a single model.

III. Methodology

To address the challenges, we present our approach, called TCAE, to dynamically generate the weights to ensemble predictions of base models for a certain traffic condition. In this section, we design a traffic flow matrix that is used as the input of a CNN model to extract spatiotemporal features. The convolution operation is intended to extract spatial features and pooling operation is intended to extract temporal features. After extracting the features of traffic flow, we use an additive model to ensemble the predictions of base models.

A. Problem Formulation

The task of traffic prediction is to learn a function that maps observed traffic data (such as traffic volume, travel time) and exogenous information (such as extreme weather) to a target traffic value. Although fusing traffic data and exogenous information is a promising approach to enhance traffic data mining, it is beyond the scope of this paper. We focus on developing the methodology to ensemble multiple models based on the spatiotemporal characteristics embedded in traffic flow. To formally define this problem, we use the notation x_t^i to represent the traffic flow of detector i at time t. As aforementioned, the spatiotemporal dependency of traffic flow, as an important feature of traffic dynamics, can contribute to traffic prediction. To demonstrate the spatiotemporal characteristics, a near-term traffic flow matrix is defined as

$$\boldsymbol{X}_t = \begin{pmatrix} x_{t-\tau+1}^1 & \cdots & x_{t-1}^1 & x_t^1 \\ x_{t-\tau+1}^2 & \cdots & x_{t-1}^2 & x_t^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_{t-\tau+1}^N & \cdots & x_{t-1}^N & x_t^N \end{pmatrix} \tag{2}$$

In this matrix, $\boldsymbol{x}_{t-\tau+1:t}^i = (x_{t-\tau+1}^i, \ldots, x_{t-1}^i, x_t^i)^{\mathrm{T}}$ is the historical $\tau - step$ series of detector i that explicitly contains temporal characteristics. Although concatenating the temporal series of N detectors does not explicitly model their spatial correlations, it would contribute to extract spatial information with following convolutional operations. And notation $\hat{y}_{t+1}^j$ is used to represent the estimated traffic value for detector j at time $t+1$, then the map function $\Re$ is written as

$$\boldsymbol{\Re} : \boldsymbol{X}_t \rightarrow \hat{y}_{t+1}^j \tag{3}$$

B. Spatiotemporal Representation Learning Using CNN

CNN, a feed forward neural network that consists of convolutional layers interspersed with pooling layers, aims to learn regional features. In this work, we intend to extract temporal features with convolution operation, and learn spatial features with pooling operation. The convolution operation can be summarized as moving a filter over the 2-Dimension traffic matrix and computing the dot products. To effectively extract temporal features, a convolution operation only overlays a feature vector $\boldsymbol{x}_{t-\tau+1:t}^i$ that represents a historical traffic series as shown in Fig. 2. Thus, the first dimension of the filter is 1, and given $1 \times V$ filter matrix $\boldsymbol{k}^{\mathrm{T}}$, the $p-th$ element of the output feature vector is

$$a_p^i = f(conv(\left[\boldsymbol{x}_{t-\tau+p:t-\tau+p+V-1}^i), \boldsymbol{k}\right]) + b) \tag{4}$$

where b is the bias and f is the activation function. And the operation $conv(\cdot)$ is defined as

$$conv(\left[\boldsymbol{x}_{t-\tau+p:t-\tau+p+V-1}^i, \boldsymbol{k}\right]) = \boldsymbol{k}^{\mathrm{T}}\boldsymbol{x}_{t-\tau+p:t-\tau+p+V-1}^i \tag{5}$$

Substituting 5 into 4, we obtain

$$a_p^i = f(\boldsymbol{k}^{\mathrm{T}}\boldsymbol{x}_{t-\tau+p:t-\tau+p+V-1}^i + b) \tag{6}$$

Applying 6 repeatedly from when p equals 1 to $(\tau - V + 1)$, we obtain the feature vector $\boldsymbol{a}^i = (a_1^i, a_2^i, \cdots, a_{\tau-V+1}^i)^{\mathrm{T}}$ of the convolution operation on feature vector $\boldsymbol{x}_{t-\tau+1:t}^i$. If there are M detectors in this case, the dimension of output map is $M \times (\tau - V + 1)$ and written as

$$\boldsymbol{A_k} = (\boldsymbol{a}^1, \cdots, \boldsymbol{a}^M)^{\mathrm{T}} \tag{7}$$

Convolution with one filter outputs one feature map. To learn more sophisticated features, generally there are distinct filters to convolve the input map, then the concatenated output maps would be passed to a pooling layer. To effectively extract spatial features, the pooling operation computes the average or maximum of each region on one column of the feature map. This operation reduces the dimension of feature map by merging neighboring points, so that computation time is reduced and more abstract information would be passed to the next layers. In this paper, to make the abstract features focuses on the dominant location or detector, maximum pooling is used to down-sample the feature map. The $q-th$ element of pooling output on feature vector $\boldsymbol{a}_j = (a_j^1, \cdots, a_j^M)^{\mathrm{T}}$ is

$$o_j^q = f(\beta \max\{a_j^q, \cdots, a_j^{q+m-1}\} + b) \tag{8}$$

where β is the multiplicative bias, b is the additive bias, m is the size of pooling window, and f is the activation function. Thus the output of pooling is written as

$$\boldsymbol{O} = \begin{pmatrix} o_1^1 & \cdots & o_{\tau-V+1}^1 \\ \vdots & \vdots & \vdots \\ o_1^{q_{out}} & \cdots & o_{\tau-V+1}^{q_{out}} \end{pmatrix} \tag{9}$$

where q_{out}.

After a series of convolution and pooling operations, the last output map contains the extracted features from raw traffic matrix, and it is flattened into a feature vector. This vector is fed into fully connected layers and the network outputs the weight vector $\left(\alpha_i \cdots \alpha_j\right)^{\mathrm{T}}$, which is used to ensemble multiple predictors.

C. Ensemble Learning for Traffic Prediction

The motivations to exploit the different performances of the base predictors on estimating the same target value, and ensemble multiple models are two-fold: (1) to enhance the accuracy, and (2) to reduce the variance. The proposed method considers an additive model in the following form

$$\begin{aligned} \hat{y}_t &= \Re(\boldsymbol{X}_{t-1}) \\ &= \sum_i \alpha_{t,i}\hat{y}_{t,i} \end{aligned} \tag{10}$$

where $\alpha_{t,i}$ is the weight learned using CNN-based spatiotemporal representation, and $\hat{y}_{t,i}$ is the output of the $i-th$ traffic predictor that is taken as a base predictor. Though the base predictors can be arbitrary models in theory, it is critical to select dependent models in practice as the diversity of base predictors makes it possible to exploit their different behaviors. Although not mandatory, the base models should have been fitted before applied to ensemble learning in this paper. Then, our proposed method focuses on tuning the parameters of CNN-based traffic condition sensing model. The loss function is defined as

$$L = L(y_t, \sum_i \alpha_{t,i}\hat{y}_{t,i}) \tag{11}$$

where $\boldsymbol{\alpha}_t = \left(\alpha_{t,i}\right)$is the output of the CNN model and is written as

$$\boldsymbol{\alpha}_t = \Psi(\boldsymbol{X}_{t-1}|\phi) \tag{12}$$

where ϕ is the trainable parameter set of CNN model. Then, the objective of learning an effective ensemble model is achieved by minimizing the loss function L. Gradient descent algorithm is used as the optimization strategy, and parameters ϕ are updated by

$$\phi = \phi - \eta\frac{\partial L(y_t, \Psi(\boldsymbol{X}_{t-1}|\phi)\hat{\boldsymbol{y}}_t)}{\partial \phi} \tag{13}$$

The process of training ensemble model for traffic prediction based on spatiotemporal representation learning is summarized in Algorithm 1.

Algorithm 1 Training Ensemble Model for Traffic Prediction

Input: traffic flow training dataset $D = \{(\boldsymbol{X}, y)\}$, learning rate η, batch size b, base models $\Gamma = (\Gamma_i)$
Output: ensemble model Ψ

```
initialize φ
while stopping criterion not met do
    sample a mini-batch of b samples from
     D: {X^(1), ··· , X^(m)} with corresponding
     targets {y^(1), ··· , y^(m)}
    for i = 1 to b do
        //forward propagation
        ŷ^(i)_base = Γ(X^(i))
        ŷ^(i) = Ψ(X^(i)|φ) ŷ^(i)_base
        //back propagation
        L = L(y^(i), ŷ^(i))
        ∇L = ∇L + ∂L(y_t, ŷ_t)/∂φ
    end for
    //update φ
    φ = φ − (1/b) η ∇L
end while
```

IV. Experiments

A. Dataset and Experiments Settings

1) Dataset Description and Data Preprocessing: Traffic flow data collected by Caltrans Performance Measurements Systems (PeMS) have been widely used for researchers to develop and evaluate traffic models, and are also used to evaluate the proposed method in this paper. Caltrans PeMS has placed over 39,000 individual detectors spanning the freeway system across all major metropolitan areas of the State of California. The 5-min traffic flow data of District 5 named Central Coast in 2013 are selected for the experiments. There are 147 vehicle detector stations (VDSs) deployed in this area. And the data from 6:00 Am to 10:00 Pm of 363 days are used as there exist null numbers in the data of other two days

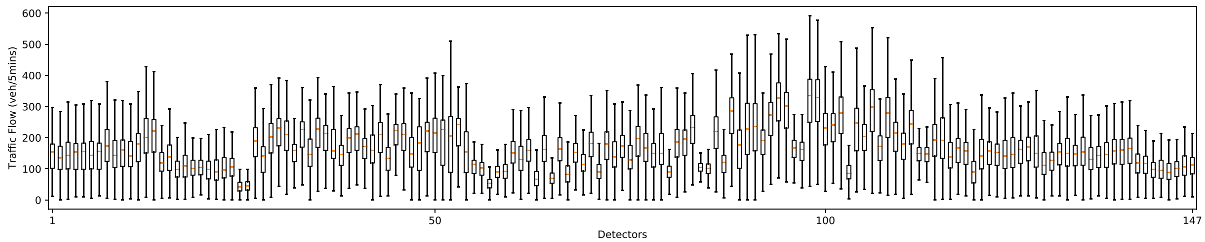


Fig. 3. Box plot of traffic flow data.

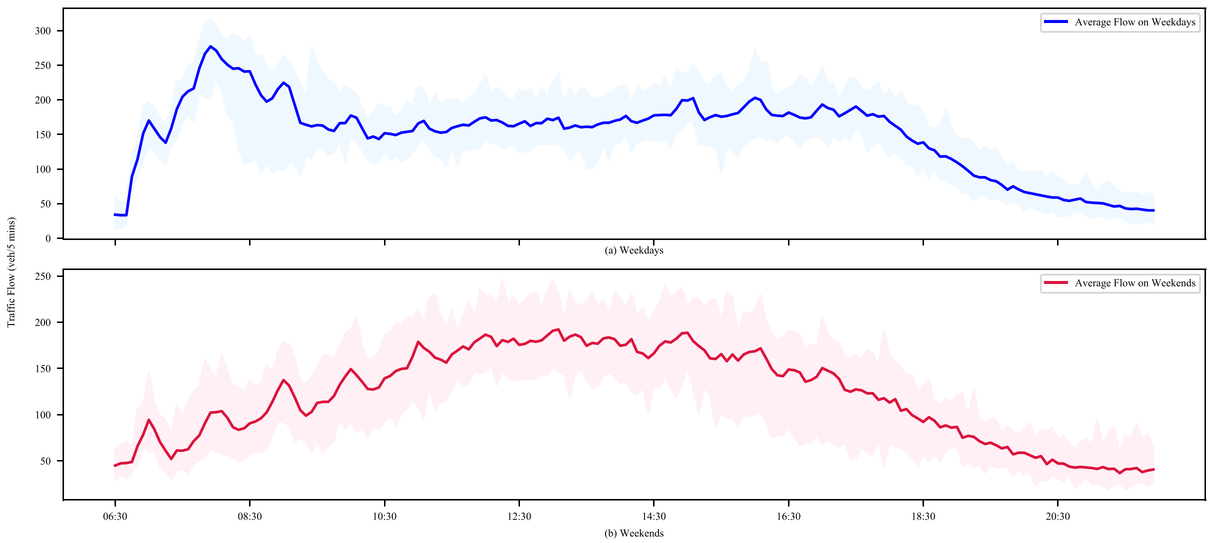


Fig. 4. Typical temporal patterns in traffic flow. The upper bound of the light-colored band is the line of maximum traffic flow and the lower bound of the light-colored band is the line of minimum traffic flow.

throughout the studied period. As shown in Fig. 3, we use boxplot to graphically depicting the spread and centers of the traffic flow dataset used in this work. For each VDS, five-number summary, i.e. the minimum, first quartile, median, third quartile, and maximum, is displayed. We use z_score normalization to transform the data of each VDS to be centered around 0 and with a standard deviation of 1. Z_score normalization is written as

$$y^* = \frac{y - \bar{y}}{\sigma} \tag{14}$$

where y is the raw data, $\bar{y}$ is the mean value of the raw data and σ is the standard deviation of the raw data. Due to limited computational resource, the first 50 VDSs are selected to evaluate the proposed method, although all data of 147 VDSs are used as input. Then, the data of the first 80 percentage are split into the training set, and the remaining data are split into the testing set.

Traffic flow data are typical spatial time series which share temporal and spatial dependencies. Fig. 4 is a plot of a typical freeway's traffic flow at weekdays and weekends. To make full use of spatiotemporal characteristics, the time series feature vector $\boldsymbol{x}_{:}^{i}$ comprises successive 6 traffic flow point, namely $(x_{t-5}^i, \ldots, x_{t-1}^i, x_t^i)^{\mathrm{T}}$ which explicitly shows the temporal dependencies of the VDS i. As the training dataset covers 147 VDSs, the traffic matrix $\boldsymbol{X}_t$, comprising the time series feature vector of all VDSs, is $(\boldsymbol{x}_{t-5:t}^1, \boldsymbol{x}_{t-5:t}^2, \cdots, \boldsymbol{x}_{t-5:t}^{147})^{\mathrm{T}}$

2) Model Configurations: Pytorch, an open source software library for numerical computation, is used to build the neural network models including LSTM, CNN and the ensemble model. And SVR, Weight Regression (WRegression) and GBRT are implemented with another machine learning library, scikit-learn. As shown in Fig. 5 the proposed model comprises one convolution layer that has 5 filters with size (1, 3), one pooling layer with kernel size (10, 1), one flatten layer and two fully connected layers. The activation functions of convolution layer and former fully connected layer are Relu. The input shape and output shape of each layer are also illustrated in Fig. 5. The Adam algorithm is used to optimize loss function of ensemble model, and the learning rate is set to 0.001.

As aforementioned, to design an effective ensemble model, the base predictors should be dependent and can make different predictions under the same condition. As reported in previous studies, HA, SVR and LSTM achieve high performance and show their different advantages under different traffic condition, which leads us to apply SVR and LSTM as the base predictors. A stacked LSTM model, comprising two LSTM layer with 6 -dimensional hidden state and one fully connected output layer, is used to learn traffic patterns. The SVR model uses radial basis function (RBF) as the kernel function, and its penalty parameter is set to empirical value 1.0. For GBRT, there are 500 estimators whose maximum depth is 4, and the learning rate is set to 0.01.

To evaluate the improvement achieved by the ensemble model based on CNN, we compare our approach with two competitive ensemble methods, including Gradient Boosting Regression Tree (GBRT) and WRegression of the prediction

TABLE I

PERFORMANCE COMPARISONS OF HA, SVR, LSTM, CNN, WREGRESSION, GBRT AND THE PROPOSED APPROACH

Method		MAE	$\pm\delta(MAE)$	MRE	$\pm\delta(MRE)$	$RMSE$	$\pm\delta(RMSE)$
Single Model Approach	SVR	13.1604	±2.7023	0.1061	±**0.0248**	17.7800	±3.4985
	LSTM	13.3999	±2.7472	0.1098	±0.0274	18.0299	±3.5550
	HA	13.9525	±2.9904	0.1102	±**0.0241**	19.0658	±3.9540
	CNN	21.9240	±9.5955	0.1791	±0.0827	30.9490	±12.0751
Ensemble Model Approach	WRegression	13.1788	±2.7023	0.1071	±0.0256	17.7886	±3.4997
	GBRT	13.2907	±2.7318	0.1095	±0.0272	17.8656	±3.5138
	TCAE(LSTM,HA)	13.0387	±**2.6921**	0.1067	±0.0271	17.4541	±3.4902
	TCAE(HA,SVR)	**12.9583**	±2.6932	**0.1045**	±**0.0243**	**17.3950**	±**3.4827**
	TCAE(SVR,LSTM)	**12.9443**	±**2.6830**	**0.1048**	±**0.0248**	**17.3748**	±**3.4681**
	TCAE(SVR,LSTM,HA)	**12.9113**	±**2.6797**	**0.1051**	±**0.0248**	**17.3086**	±**3.4779**

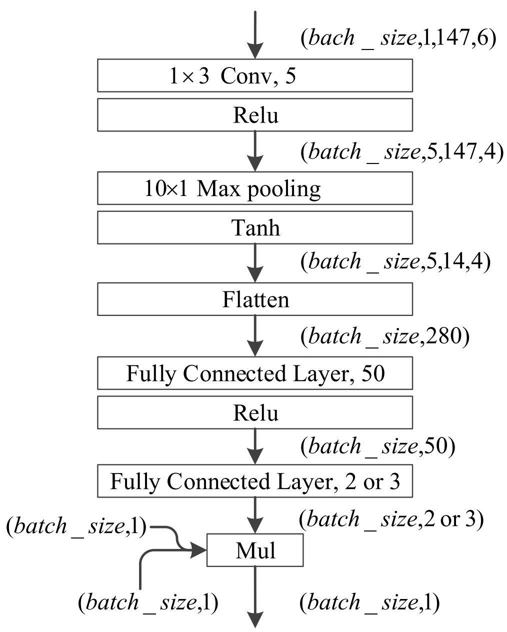


Fig. 5. An implementation of ensemble model using CNN.

of base models. GBRT has been used in a variety of areas including traffic prediction and achieved superior performance. WRegression method simply combines multiple outputs of base predictors, and also proves to be effective. The prediction $\hat{y}$ of WRegression is formally written as

$$\hat{y} = \sum_{i} w_i \hat{y}_i \tag{15}$$

where $\hat{y}_i$ is the prediction of base model i, and w_i is its weight. The objective is to minimize the error between target y and its estimation $\hat{y}$ by adjusting weights.

Besides ensemble methods, we also compare the proposed method with the selected base predictors, i.e., SVR, LSTM and HA. As our ensemble approach applies CNN to capture the spatiotemporal characteristics embedded in traffic matrix, the same CNN architecture, except that the output of last fully connected layer is changed from $(batch\_size, 2)$ to $(batch\_size, 1)$, is used to make predictions directly and its performance is compared with that of the proposed method.

B. Evaluation Metrics

To evaluate the performances of the proposed traffic prediction approach, we employ six statistical metrics, which are Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Relative Error (MRE), and the standard deviations. MAE, MRE and $RMSE$ are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \left|y_i - \hat{y}_i\right| \tag{16}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N} \frac{\left|y_i - \hat{y}_i\right|}{y_i} \tag{17}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{18}$$

where y_i is the target value, $\hat{y}_i$ is the prediction, and N is the number of instances in test dataset. The standard deviations of MAE, MRE and $RMSE$ are defined as

$$\delta(MAE) = \sqrt{\frac{\sum_{s=1}^{S}(MAE_s - \overline{MAE})^2}{S-1}} \tag{19}$$

$$\delta(MRE) = \sqrt{\frac{\sum_{s=1}^{S}(MRE_s - \overline{MRE})^2}{S-1}} \tag{20}$$

$$\delta(RMSE) = \sqrt{\frac{\sum_{s=1}^{S}(RMSE_s - \overline{RMSE})^2}{S-1}} \tag{21}$$

where S is the number of vehicle detector stations, MAE_s, MRE_s and $RMSE_s$ is the MAE, MRE and $RMSE$ for vehicle detector station s, and $\overline{MAE}$, $\overline{MRE}$ and $\overline{RMSE}$ is the average of MAE_s, MRE_s and $RMSE_s$, respectively.

C. Results and Discussions

1) Qualitative and Quantitative Performances: The qualitative results are partly illustrated in Fig. 6, which also include the observed traffic flow for comparison. Overall, the

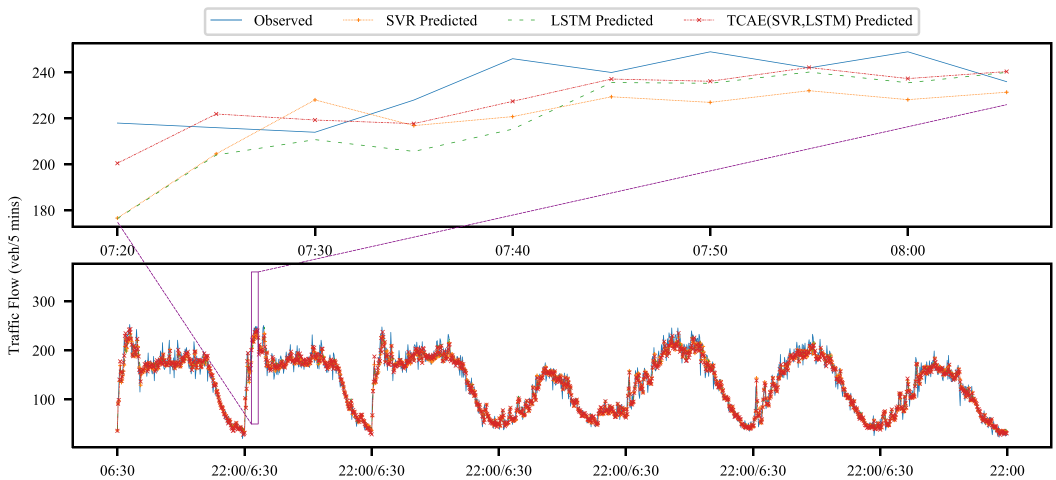


Fig. 6. Traffic prediction of based models and ensemble model over a week.

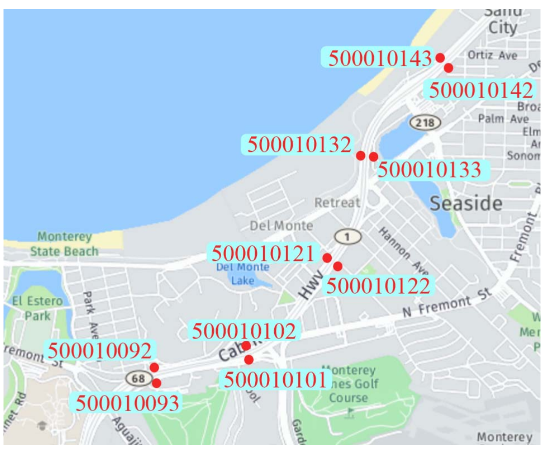


Fig. 7. The road network and the locations of ten vehicle detector stations.

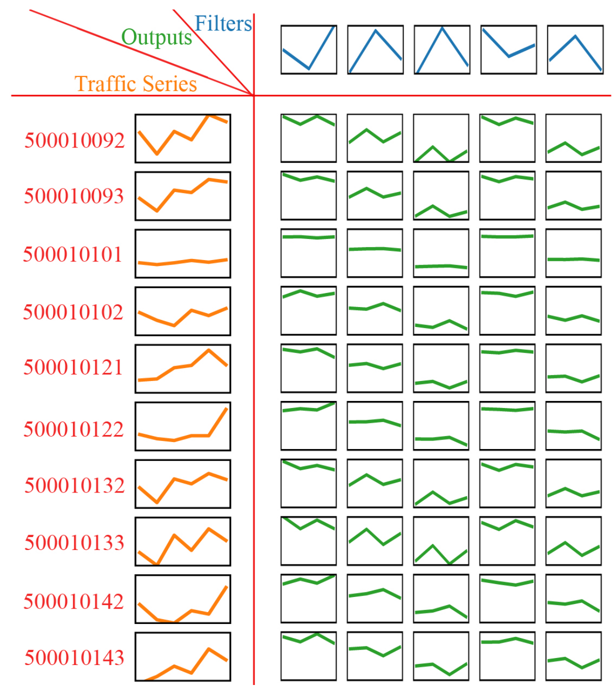


Fig. 8. Illustration of the temporal features extracted by convolution operation.

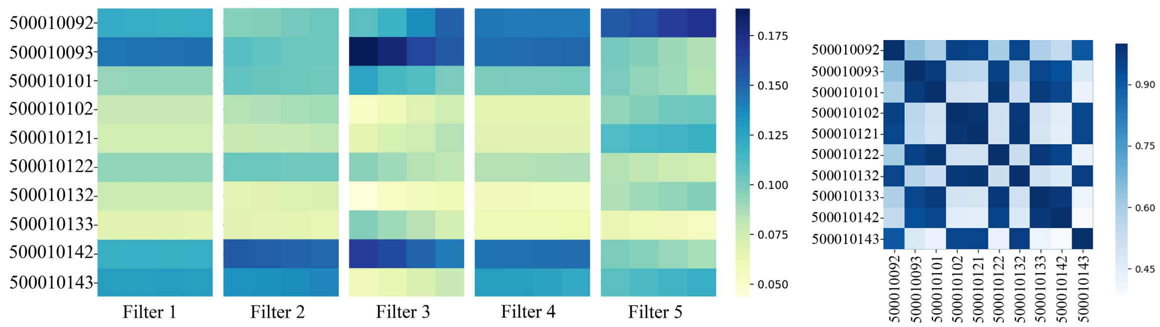
prediction error of SVR is lower than that of LSTM model, and the ensemble method proposed achieved better performances than its base models did. It is worth to mention that ensemble method does not simply sum the outputs of base models, but have the ability to learn the weights of base models according to traffic conditions. For example, the predictions of base models at 7:40 are both smaller than the observed data, however, the ensemble approach outputs a higher prediction, which is closer to observed ground-truth data and bigger than that of both base models.

The quantitative experimental results are listed in Table I. The top three scores of prediction error on the whole 50 stations along with standard deviations of errors are shown in bold in this table. Results show that the proposed approach achieved the best performances under almost all criteria.

To examine the effectiveness of ensemble paradigm, we compare the performances of base models and TCAE. There are three base models in our experiments, thus there are four combinations, i.e., SVR+HA, SVR+LSTM, HA+LSTM and SVR+LSTM+HA, to be applied in TCAE. It is clear that our approach can make significant improvements for all combinations.

In addition, we compare our approach with other competitive ensemble methods, i.e., WRegression and GBRT. Results in Table I show four TCAE approaches all outperform the competitive methods.

2) Benefit of Real-Time Traffic Condition Sensing: The proposed model and WRegression model are ensemble approaches that simply combine the estimations of base predictors, i.e., SVR and LSTM, and yield significant improvements. The results give convincing evidence that an effective ensemble model can take the benefits of different base models. As for the differences between these two models, WRegression approach only focuses the estimations of base predictors and tries to minimize the sum of prediction error in statistics, while the proposed model learns from the traffic condition and corresponding estimation pair. In other words, the proposed approach learns spatiotemporal characteristics and then makes a decision on choosing the weight values. When this decision goes into effect, the optimization algorithm will check



(a) Frequency of a point selected by max pooling (b) Pearson correlation coefficient of traffic series

Fig. 9. Illustration of spatial dependency.

the estimation errors and update the model. This procedure is repeated, and the ensemble model learns from historical experiences and becomes "smarter". It is clear the proposed method outperforms the WRegression method according to the experimental results, and we think the extra traffic-condition-awareness paradigm benefits ensemble approach.

3) Benefit of Spatiotemporal Features Extraction by Convolution and Max Pooling: As aforementioned, learning extra traffic condition in ensemble model plays an important role and helps improve model performance. In this paper, we use CNN to extract the spatiotemporal features embedded in traffic matrix. Thus, we trained a CNN model with the same architecture in the proposed ensemble model as a traffic predictor. The results show that CNN predictor only outperforms the SVR model under MAR and MRE criteria. This indicates that CNN model is good at sensing the abstract and global traffic conditions compared with directly predicting precise and local traffic flow values.

To further show how the convolution and max pooling operations help the TCAE achieve better performances, we visualize the features extracted. In our experiments, the data of all 147 VDSs are used as input, but for clarity, we only select ten VDSs, as shown in Fig. 7, to depict the spatiotemporal features extracted.

We plot the filters, inputs, and outputs of convolution operation in Fig. 8. There are five filters in our experiments and each filter can be regarded as a certain pattern. As the convolution operation can be summarized as moving a filter over the 2-Dimension traffic matrix and computing the dot products. Thus, whenever the filter comes across a pattern like that in the traffic matrix, it gives a high output. Taking VDS 500010092 as an example, we see that the first filter looks like a "V" shape, and the first position to be convolved has a similar shape while the second position not, thus the convolution output of the first one is bigger than that of the second one. In short, the outputs of convolution convey the temporal patterns and higher outputs means the traffic series are more similar to the filters.

Next the temporal features are used as inputs for max pooling operation, which aims to extract the spatial features. In our experiments, the shape of temporal features is $(147, 4)$. The max pooling operation is to take the max of a window of temporal features along the first axis. The points not selected by max pooling are neglected in following computation, and each point selected by max pooling plays an important role on the model. Fig. 9(a) shows the frequency of each point selected by max pooling for predicting the traffic of VDS 500010122. From Fig. 9(a), we find that:

1. The points of VDS 500010093 are selected frequently in terms of the third filter, especially the first ones, which indicates the importance of these points for predicting the traffic of VDS 500010122. The situations of VDS 500010142 are similar to VDS 500010093. Looking back at Fig. 7, we find that they are the upstream and downstream stations of the target VDS 500010122. We know that the upstream and downstream detectors benefit traffic prediction in previous studies which has been confirmed in our experiments.
2. The nearest upstream and downstream stations of the target station is VDS 500010101 and VDS 500010133, respectively. However, the points of these two stations are only selected very few times. The information contained in the nearest stations may be redundant for predicting the target station if they are too close.
3. The points of target VDS 500010122 are also selected very few times. In our experiments, base models have already extracted the patterns embedded in the target station, and we think the base models are good enough. Thus, there are no needs to pay more attention to itself. To improve prediction performances, the ensemble model should find some useful information conveyed in the data of other spatially dependent stations. This finding is in accordance with our expectation.

In addition, we show the Pearson correlation coefficient of traffic series in Fig. 9(b). The Pearson correlation coefficient is used to measure the strength of a linear association between two variables, where the value $r = 1$ means a perfect positive correlation and the value $r = -1$ means a perfect negative correlation. From Fig. 9(b), we see that the degree of correlation between two stations in the same direction is high. Therefore, high correlation coefficient only indicates when one increases, the other increases, and it does not assure that there is a cause and effect relationship. Besides, it is worthy to mention that the points of VDS 50001092 are also selected frequently, which is hard to explain as the neural network is a "black box". In the future, we will do more studies on the interpretability of neural network models.

V. Conclusion

Although ensemble learning can help improve the performances of traffic prediction, a fixed ensemble rule is not the best way to combine the predictions of base models. In this paper, we design a novel ensemble strategy that adaptively generate the weights to ensemble base models by evaluating their performances under different traffic conditions. To learn the relationships between model performances and traffic patterns, we proposed to apply a CNN model to learn spatiotemporal patterns embedded in traffic flow, and further to utilize the abstract features extracted by CNN to ensemble multiple models. We evaluated the performances of the proposed approach on traffic flow data from Caltrans PeMS, and compare it with competitive methods, i.e., Support Vector Regression Model, Long Short-Term Memory Model, Historical Average Model, Weight Regression Model and Gradient Boosting Regression Tree Model. Experimental results showed that the proposed method leads to an improvement in traffic flow data prediction. In addition, the ensemble rule of WRegression model is similar to our approach except directly sensing extra traffic patterns and adaptively generating the ensemble weights. It is clear the proposed two-stage decision making approach outperforms the WRegression method. Furthermore, we use convolution and max pooling to detect the traffic patterns and the visualization of these layers shows the different effects of different sensors on the prediction of a certain sensor. In short, the nearest sensors and the target sensor have less effect in the ensemble learning than further upstream and downstream sensors as the traffic flow patterns in the former sensors is redundant. We further calculate the correlation coefficient of these sensors, and find high correlation coefficient does not assure that there is a cause and effect relationship.

In future research, the interpretability of neural network model is one main direction to explore in the decision making framework as the neural network is a "black box" and some middle results are hard to understand and explain. Besides, the using of CNN model with convolution and pooling operations to extract spatiotemporal features is non-intuitive and hard to explain, we plan to investigate other effective neural networks to detect the spatiotemporal features embedded in traffic flows, such as graph convolutional network. To demonstrate the effectiveness of the two-stage decision making framework, we plan to apply our approach to more other tasks in this field, such as travel time prediction and object classification.

References

[1] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[2] L. Li, X. Chen, and L. Zhang, "Multimodel ensemble for freeway traffic state estimations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1323–1336, Jun. 2014.

[3] X. Chen, S. Zhang, and L. Li, "Multi-model ensemble for short-term traffic flow prediction under normal and abnormal conditions," *IET Intell. Transp. Syst.*, vol. 13, no. 2, pp. 260–268, Feb. 2019.

[4] X. Ma, C. Ding, S. Luan, Y. Wang, and Y. Wang, "Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2303–2310, Sep. 2017.

[5] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[6] Y. Chen, Y. Lv, and F. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1624–1630, Apr. 2020.

[7] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[8] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[9] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, Feb. 1984.

[10] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: A survey," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 24–38, May 2009.

[11] Y. Zhao-Sheng, W. Yuan, and G. Qing, "Short-term traffic flow prediction method based on SVM," *J. Jilin Univ. (Eng. Technol. Ed.)*, vol. 6, no. 36, p. 009, 2006.

[12] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[13] J. Arroyo and C. Maté, "Forecasting histogram time series with K-nearest neighbours methods," *Int. J. Forecasting*, vol. 25, no. 1, pp. 192–207, Jan. 2009.

[14] B. Yu, X. Song, F. Guan, Z. Yang, and B. Yao, "K-nearest neighbor model for multiple-time-step prediction of short-term traffic condition," *J. Transp. Eng.*, vol. 142, no. 6, Jun. 2016, Art. no. 04016018.

[15] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9508–9517, Dec. 2016.

[16] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[17] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734.

[18] F. A. Gers, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Networks: ICANN*, vol. 2, 1999, pp. 850–855.

[19] R. Li, Y. Hu, and Q. Liang, "T2F-LSTM method for long-term traffic volume prediction," *IEEE Trans. Fuzzy Syst.*, early access, Apr. 13, 2020, doi: 10.1109/TFUZZ.2020.2986995.

[20] Y. Liu, J. J. Q. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7751–7763, Aug. 2020.

[21] D. Ma, X. Song, and P. Li, "Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter-and intra-day traffic patterns," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 24, 2020, doi: 10.1109/TITS.2020.2973279.

[22] T. Jia and P. Yan, "Predicting citywide road traffic flow using deep spatiotemporal neural networks," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 16, 2020, doi: 10.1109/TITS.2020.2979634.

[23] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, early access, May 21, 2020, doi: 10.1109/TITS.2020.2983763.

[24] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "TrafficGAN: Network-scale deep traffic prediction with generative adversarial nets," *IEEE Trans. Intell. Transp. Syst.*, early access, Dec. 17, 2019, doi: 10.1109/TITS.2019.2955794.

[25] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert Syst. Appl.*, vol. 38, no. 1, pp. 223–230, Jan. 2011.

[26] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[27] Y. Su, Y. Zhang, D. Ji, Y. Wang, and H. Wu, "Ensemble learning for sentiment classification," in *Proc. Workshop Chin. Lexical Semantics*. Cham, Switzerland: Springer, 2012, pp. 84–93.

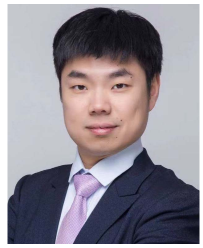
[28] G. Liang, X. Zhu, and C. Zhang, "An empirical study of bagging predictors for different learning algorithms," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 1802–1803.
[29] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation Classification*. Cham, Switzerland: Springer, 2003, pp. 149–171.
[30] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1915–1919.
[31] T. G. Dietterich, "Ensemble methods in machine learning," in *Proc. 1st Int. Workshop Multiple Classifier Syst.*, vol. 1857, no. 1, 2000, pp. 1–15.
[32] L. Chen and C. L. P. Chen, "Ensemble learning approach for freeway short-term traffic flow prediction," in *Proc. IEEE Int. Conf. Syst. Syst. Eng.*, Apr. 2007, pp. 1–6.
[33] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 308–324, Sep. 2015.
[34] S. Yang, J. Wu, Y. Du, Y. He, and X. Chen, "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *J. Sensors*, vol. 2017, pp. 1–15, May 2017.
[35] J. Xiao, Z. Xiao, D. Wang, J. Bai, V. Havyarimana, and F. Zeng, "Short-term traffic volume prediction by ensemble learning in concept drifting environments," *Knowl.-Based Syst.*, vol. 164, pp. 213–225, Jan. 2019.
[36] C. Ding, D. Wang, X. Ma, and H. Li, "Predicting short-term subway ridership and prioritizing its influential factors using gradient boosting decision trees," *Sustainability*, vol. 8, no. 11, p. 1100, Oct. 2016.
[37] F. Zhang, X. Zhu, T. Hu, W. Guo, C. Chen, and L. Liu, "Urban link travel time prediction based on a gradient boosting method considering spatiotemporal correlations," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 11, p. 201, Nov. 2016.
[38] F. Zhao, G.-Q. Zeng, and K.-D. Lu, "EnLSTM-WPEO: Short-term traffic flow prediction by ensemble LSTM, NNCT weight integration, and population extremal optimization," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 101–113, Jan. 2020.
[39] J. Wang, W. Deng, and Y. Guo, "New Bayesian combination method for short-term traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 79–94, Jun. 2014.
[40] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An improved Bayesian combination model for short-term traffic prediction with deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1332–1342, Mar. 2020.

Yuanyuan Chen (Member, IEEE) received the B.E. degree in automation from Tongji University, Shanghai, China, in 2013, and the Ph.D. degree in control theory and control engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2018.

He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. His research interests include social transportation systems and data-driven traffic modeling and prediction.

Hongyu Chen received the B.E. degree in automation from Chang'an University, Xi'an, China, in 2017. She is currently pursuing the M.E. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China.

Her research interests include traffic data analysis, social transportation, and intelligent transportation.



Peijun Ye (Member, IEEE) received the Ph.D. degree from the University of Chinese Academy of Sciences in 2013.

He was a Visiting Scholar with the Department of Cognitive Science, University of California at San Diego, from 2017 to 2018. He is currently an Assistant Research Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include multiagent systems, artificial intelligence, social simulation, and intelligent transportation systems. He has authored/coauthored over 30 academic articles, five invention patents, four software copyrights, and one monograph. He serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, and a Secretary for the Social and Economic Computing ACM Chapter. He is also an Invited Reviewer of *Journal of Artificial Societies and Social Simulation*, *Knowledge and Information Systems*, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA, and the IEEE International Conference on Intelligent Transportation Systems (2011–2019).

Yisheng Lv (Senior Member, IEEE) received the B.E. and M.E. degrees in transportation engineering from the Harbin Institute of Technology, Harbin, China, in 2005 and 2007, respectively, and the Ph.D. degree in control theory and control engineering from the Chinese Academy of Sciences in 2010.

He is currently an Associate Professor with the Institute of Automation, Chinese Academy of Sciences. His research interests include traffic data analysis, dynamic traffic modeling, and parallel traffic management and control systems.

Fei-Yue Wang (Fellow, IEEE) received the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

In 1990, he joined The University of Arizona and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program from the State Planning Council, and in 2002, he was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems. His current research interests include the methods and applications for parallel intelligence, social computing, and knowledge automation. He was elected as a fellow of INCOSE, IFAC, ASME, and AAAS. In 2007, he received the National Prize in Natural Sciences of China and became an Outstanding Scientist of ACM for his work in intelligent control and social computing. He received the IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, and the IEEE SMC Society Norbert Wiener Award in 2014. Since 1997, he has been serving as the General or Program Chair for more than 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, the Chinese Association for Science and Technology, USA, in 2005, the American Zhu Kezhen Education Foundation from 2007 to 2008, the Vice President of the ACM China Council from 2010 to 2011, the Vice President and the Secretary General of the Chinese Association of Automation from 2008 to 2018. He was the Founding Editor-in-Chief (EiC) of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, the *IEEE Intelligent Transportation Systems Magazine* from 2006 to 2007, the IEEE/CAA JOURNAL OF AUTOMATICA SINICA from 2014 to 2017, and the China's *Journal of Command and Control* from 2015 to 2020. He was the EiC of the IEEE INTELLIGENT SYSTEMS from 2009 to 2012, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS from 2009 to 2016, and is the EiC of the IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS since 2017, and the Founding EiC of *Chinese Journal of Intelligent Science and Technology* since 2019. He is currently the President of CAA's Supervision Council, IEEE Council on RFID, and the Vice President of IEEE Systems, Man, and Cybernetics Society.