DOI: 10.1111/mice.12459

ORIGINAL ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

A spatio-temporal ensemble method for large-scale traffic state prediction

Yang Liu[1] | Zhiyuan Liu[1] | Hai L. Vu[2] | Cheng Lyu[1]

[1]Jiangsu Key Laboratory of Urban ITS, Jiangsu Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, Nanjing, 211189, China

[2]Department of Civil Engineering, Institute of Transport Studies, Monash University, Clayton, Victoria 3800, Australia

Correspondence
Zhiyuan Liu, Jiangsu Key Laboratory of Urban ITS, Jiangsu Collaborative Innovation Center of Modern Urban Traffic Technologies, School of Transportation, Southeast University, Nanjing, 211189, China.
Email: zhiyuanl@seu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: General Projects (No. 71771050), Key Projects (No. 51638004); Fundamental Research Funds for the Central Universities, Grant/Award Number: 2242018K41023/2242019K41012; Scientific Research Foundation of Graduate School of Southeast University, Grant/Award Number: YBPY1927

Abstract

How to effectively ensemble multiple models while leveraging the spatio-temporal information is a challenging but practical problem. However, there is no existing ensemble method explicitly designed for spatio-temporal data. In this paper, a fully convolutional model based on semantic segmentation technology is proposed, termed as spatio-temporal ensemble net. The proposed method is suitable for grid-based spatio-temporal prediction in dense urban areas. Experiments demonstrate that through spatio-temporal ensemble net, multiple traffic state prediction base models can be combined to improve the prediction accuracy.

1 | INTRODUCTION

Real-time traffic state prediction is an essential component for traffic control and management in an urban road network. It enables the identification and monitoring of congestion while the information assists in the implementation of advanced traveler information systems (Jiang & Adeli, 2004). Recently, applications of traffic state prediction in various transportation scenarios have been observed, for example, a smart driving intelligence system (Yuan, Zheng, Xie, & Sun, 2013), a taxi-sharing system (Ma, Zheng, & Wolfson, 2013), and an integrated corridor traffic management (Hashemi & Abdelghany, 2016). Nevertheless, due to technological and financial limitations, it is still a challenging task to obtain city-wide noise-free data containing multiple traffic parameters such as traffic volume, speed, and density. Therefore, there is an imperative need to develop accurate and proactive traffic state prediction methods for the optimization of traffic control and management measures.

Regarding data source, most past research is based on data collected from fixed sensors like loop detectors. However, several limitations exist in the installation and maintenance of these fixed sensors, including high installation cost, malfunctioning rate, and repairing difficulty. In

comparison, mobile sensors like GPS trackers show greater potential in providing researchers and operators with large-scale raw data of traffic dynamics. In recent years, advances in the internet economy have accelerated the development of urban transportation system while breeding the market of online car-hailing services. As an upgrade of traditional taxis, the online car-hailing platforms have gained significant popularity and have greatly extended the size and scope of traffic databases, facilitating the studies on large-scale traffic state prediction.

In existing works of traffic prediction, common methods can be classified into three categories, namely, traditional time series models, analytical models, and machine learning models. Thanks to the development in data collection tools, the abundance of data nowadays can facilitate a variety of studies, including traffic safety (Gu, Abdel-Aty, Xiang, Cai, & Yuan, 2019; Guo, Liu, Wu, & Chen, 2018; Liu, Wu, Zhou, Bao, & Yang, 2019), traffic management (Hashemi & Abdelghany, 2018; Li, Zhang, Wang, & Ran, 2018; Liu, Jia, Xie, & Liu, 2019; Liu, Wang, Zhou, & Cheng, 2017), energy consumption (Pan, Chen, Qiao, Ukkusuri, & Tang, 2019), and flow prediction (Liu, Liu, & Jia, 2019). Notably, the emerging deep learning methods significantly extend the possibility of more creative research, such as infrastructure monitoring (Nabian & Meidani, 2018; Rafiei & Adeli, 2017, 2018; Xue & Li, 2018) and material property examination (Rafiei, Khushefati, Demirboga, & Adeli, 2017). These studies have provided substantial contribution to the field and lay a foundation for further applications of deep learning techniques.

However, most of traffic predictions in literature were achieved only using a single model. There has been little research on how to efficiently ensemble these models to improve their prediction accuracy. Furthermore, current ensemble methods are not explicitly designed to deal with spatio-temporal data. How to effectively ensemble multiple models while leveraging the spatio-temporal information remains a challenging but practical problem.

In this paper, we investigated the problem of ensemble method and proposed a grid-based spatio-temporal data ensemble method that can be applied to two typical scenarios. First, in the case of online car-hailing apps like Didi and Uber, large-scale grid-based predictions are performed every day, and even every hour, where various complicated models with similar prediction accuracy are involved. The method proposed in this paper enables effective ensemble of the prediction results from these models. Second, with regards to other grid-based data, like fine-grained air quality and weather records, the prediction accuracy can also be improved. Moreover, this paper integrates the computer vision and intelligent transportation applications, as the innovative thoughts and frameworks in semantic segmentation and object detection may have the potential to solve traditional traffic problems.

Zhang, Zheng, Qi, Li, and Yi (2016) designed a prediction model for spatio-temporal data based on deep learning. However, they focused on a single model, which is regarded as the base learner in our traffic state prediction module. The major contribution of this paper is to propose a framework specifically designed for large-scale grid-based traffic prediction, which can achieve a higher prediction accuracy through the ensemble of the prediction results from multiple base models.

The remainder of the paper is structured as follows. In Section 3, the problem description, definitions, and notations of traffic state prediction are elaborated. In Section 4, we introduce the convolutional neural networks, as prerequisites. In Section 5, we analyze the dataset and the traffic state. Then, the novel ensemble framework is proposed in Section 6. In Section 7, the numerical experiments are discussed. Finally, the conclusions and recommendations for future work are discussed in Section 8.

2 | LITERATURE REVIEW

2.1 | Prediction of the traffic state

Most previous studies on traffic state prediction are model driven and the data used are usually generated by fixed sensors such as loop detectors and video cameras, set up along the highway.

Traditional methods generally model traffic flow as time series while conventional time series models like autoregressive integrated moving average (ARIMA) model are often employed. It was first introduced by Ahmed and Cook (1979) to conduct traffic volume forecasting. Consequently, numerous variants have been formulated by researchers, including seasonal ARIMA (SARIMA), space-time ARIMA, and vector ARIMA (Ghosh, Basu, & O'Mahony, 2009; Kamarianakis & Prastacos, 2003; Williams, Durvasula, & Brown, 1998).

In recent times, machine learning methods have gained significant popularity and are thus often used in this task as they can automatically learn the pattern of traffic dynamics. General practices include probabilistic graphical model (Antoniou, Koutsopoulos, & Yannis, 2013; Qi & Ishak, 2014; Zheng & Su, 2016) and k-nearest neighbor method (Oh, Byon, & Yeo, 2016). To better capture the nonlinear relationship in the evolution of traffic (Smith & Demetsky, 1994, 1997), the potential of neural networks was also investigated (e.g., Allström et al., 2016; Zhang, 2000). Hybrid models like the combination of neural network and wavelets analysis have also been studied extensively in predicting traffic flow (Adeli & Ghosh-Dastidar, 2004; Adeli & Jiang, 2009; Adeli & Karim, 2005; Jiang & Adeli, 2005).

The applications of deep learning techniques with fixed sensor data usually focus on traffic flow prediction. To achieve this goal, Huang, Song, Hong, and Xie (2014) proposed a deep architecture comprising a deep belief network and a multitask regression layer. Based on data collected by detector stations in California, Lv, Duan, Kang, Li, and Wang (2014) and Duan, Lv, Liu, and Wang (2016) used a deep stacked autoencoder model to impute and predict freeway traffic flow. Ma, Tao, Wang, Yu, and Wang (2015) used long short-term memory (LSTM) neural network to predict traffic speed based on data collected by microwave sensors. Polson and Sokolov (2017) combined a linear model with a sequence of tanh layers to forecast traffic flow.

Data sources like loop detector have been extensively used in past literature due to its high recognition rate and accuracy (Bertini & Leal, 2005; Dharia & Adeli, 2003; Van Lint & Hoogendoorn, 2010; Mo, Li, & Zhan, 2017; Zhang, He, Wang, & Zhan, 2015). Nevertheless, there is a high possibility that the sensor devices may malfunction due to various reasons like wearing out of mechanical devices and pavement failure. Further, these devices also suffer from noises contained in traffic data (Zheng & Su, 2016). Moreover, the deployment and maintenance of the sensor devices require substantial long-term investment, making it unrealistic to install them on every road within urban road networks. Therefore, in most cases, the sensors are only installed at few locations within the urban road network. Therefore, this problem has significantly hindered us from obtaining adequate information for the real-time traffic state prediction, which requires us to resort to other data sources. Data collected from GPS devices installed on probe vehicles, regarded as mobile sensors, are one of the most suitable alternatives to fixed detector data because they are much easier to implement and are less likely to suffer from malfunctions. The data are generated by each moving vehicle while the trajectory information like real-time position, velocity, and direction are periodically reported and recorded. The advances in communication technology and innovative business models, especially online taxi-hailing services, have made a large quantity of mobile sensor data available, whereby many researchers have shifted their focus to these mobile sensors.

Studies based on mobile sensor data are mostly data driven. The methods used for fixed sensor data, like probabilistic graphical models, are widely used in the case of mobile sensor data as well. For instance, Hofleitner, Herring, Abbeel, and Bayen (2012) presented a dynamic Bayesian network approach for traffic condition estimation and evaluated the model with probe vehicle data in San Francisco. Ramezani and Geroliminis (2012) applied Markov chains to estimate the probability distribution of arterial route travel time. Apart from these probabilistic graphical models, many other machine learning methods have also been applied to this problem, including artificial neural networks (Fusco, Colombaroni, Comelli, & Isaenko, 2015), regression tree (Wang, Cao, Xu, & Li, 2016), and support vector machine (SVM) model (Yao et al., 2017).

In the case of deep learning methods, recurrent neural network (RNN) is preferred for tasks associated with sequential data like speech recognition (Mikolov, Karafiát, Burget, Černocký, & Khudanpur, 2010), machine translation (Sutskever, Vinyals, & Le, 2014), and traffic prediction. Ma, Yu, Wang, and Wang (2015) proposed an architecture comprising of Restricted Boltzmann Machine (RBM) and RNN and applied it to predict congestion evolution. Wang, Gu, Wu, Liu, and Xiong (2016) designed an error-feedback recurrent convolutional network structure. By introducing the error-feedback neurons, the proposed network was capable of reacting to the sudden variations caused by peak flow and emergency. Notably, an influence function was designed to help recognize congestion sources. Two RNN models, namely, LSTM and gated recurrent units, were used by Fu, Zhang, and Li (2016) to predict short-term traffic flow. It should be noted that the data of these online hailing vehicles contain both spatial and temporal features that are often neglected in traffic state prediction, leading to a waste of valuable information. Niu, Zhu, and Zhang (2014) adopted RBM and SVM to learn temporal-spatial traffic flow feature for prediction. Few studies on crowd flow prediction are worth mentioning. Zhang et al. (2016) designed a prediction model for spatio-temporal data based on deep learning. The model consists of a spatio-temporal component and a global component to combine the information related to spatial dependencies, temporal closeness, and other global factors. Based on this model, Zhang, Zheng, and Qi (2017) further presented a deep spatio-temporal residual network approach to formulate the spatial properties of the flow. Yao et al. (2018) used the spatial and temporal features to form a deep spatio-temporal network framework on a multi-view basis to forecast taxi demand.

Few researchers have also attempted to fuse the data from both fixed sensors and moving vehicles. Kalman filtering technique is one of the popular models used in this direction. For example, Nanthawichit, Nakatsuji, and Suzuki (2003) integrated the probe data with detector data and studied the traffic state with a macroscopic traffic flow model. Yuan, Van Lint, Van Wageningen-Kessels, and Hoogendoorn (2014) presented a Lagrangian traffic state estimator for traffic state estimation using an extended Kalman filtering (EKF). EKF was also adopted by Nantes, Ngoduy, Bhaskar, Miska, and Chung (2016) to leverage heterogeneous data from loop detectors, Bluetooth and GPS devices. Few studies have also employed deep learning methods. For example, Deng, Shahabi, Demiryurek, and Zhu (2017) abstracted all the points where data were collected as traffic sensors and proposed a multi-task learning framework to identify and forecast traffic condition simultaneously.

Another category of methods for depicting traffic dynamics is the analytical model. One branch of these models is macroscopic traffic assignment models, the most classical of which is the four-step model (Sheffi, 1984), which has been applied in many transportation planning tools like TransCAD, VISUM, and EMME. The issue of dynamic traffic assignment (DTA) has been a subject of substantial research and debate (Liu et al., 2017; Lu & Zhou, 2014; Song, Han, Wang, Friesz, & del Castillo, 2017). The other branch is traffic flow models. Cell transmission model (CTM) is one of the representative models for traffic prediction, which is based on the Lighthill–Whitham–Richards (LWR) model (Daganzo, 1997). Szeto, Ghosh, Basu, and O'Mahony (2009) integrated the CTM with a SARIMA model to form a short-term space-time traffic forecasting strategy. Sumalee, Zhong, Pan, and Szeto (2011) used loop detector data and extended the CTM framework as stochastic CTM. Further, it was extended by Pan, Sumalee, Zhong, and Indra-payoong (2013) to incorporate the spatio-temporal correlation of traffic flow. Some studies (e.g., Han, Piccoli, & Friesz, 2016; Lo & Szeto, 2002) contain macroscopic traffic flow model in the DTA framework. Nevertheless, the analytical methods require strong assumption and complex optimizing skills. Hence, they are not as competitive and popular as data-driven methods in the application of traffic prediction, especially state-of-the-art deep learning methods.

As mentioned above, most of the past studies use only a single model for prediction. It can be challenging but a valuable task to fuse the results of individual models proposed by researchers from different domains with the aim to achieve better prediction accuracy.

2.2 | Ensemble learning

Through the ensemble learning methods, multiple outputs of weak learners can be combined to construct a superior learner with better performance (Dietterich, 2000). Common ensemble methods include bagging, blending, and stacking (Breiman, 1996; Kuncheva, 2004; Wolpert, 1992).

Rodriguez, Kuncheva, and Alonso (2006) proposed a method with the ensemble of classifiers called rotation forest, which includes splitting the features and then conducting the principal component analysis. To enhance the accuracy of weather prediction, Williams, Neilley, Koval, and McDonald (2016) incorporated spatial-temporal neighborhood bias information and used it to formulate a constraint-regularized regression problem. Zhang and Wang (2016) presented a multi-context network, with one network averaging the output of multiple DNNs and the other stacking them together. Guzman, El-Haliby, and Bruegge (2015) compared the performance of four machine learning methods and their ensembles in classifying app reviews. The ensembles outperformed individual models, exhibiting a better overall performance. Singh, Hoiem, and Forsyth (2016) proposed a stochastic learning method that sampled from a rich set of architecture and gave satisfactory results without constructing very deep networks.

However, the ensemble methods are not explicitly designed to handle spatio-temporal data. As trajectory data inherently possess both temporal and spatial attributes, caution should be taken on the use of domain knowledge while extracting relevant features as well as developing forecasting models that incorporate spatio-temporal information.

2.3 | Semantic segmentation

Deep learning has made significant progress both in the classification of the whole image (Krizhevsky, Sutskever, & Hinton, 2012; Simonyan & Zisserman, 2014) as well as on local tasks, including object detection (Cha, Choi, & Büyüköztürk, 2017; He, Zhang, Ren, & Sun, 2014; Lin, Nie, & Ma, 2017; Sermanet et al., 2014), critical point forecasting (Long, Zhang, & Darrell, 2014), and local correspondence (Fischer, Dosovitskiy, & Brox, 2014).

The coarse-to-fine inference approach involves predicting at each pixel. Semantic segmentation, the foundation of image understanding, has been widely applied to a variety of scenarios (e.g., Street View and Remote Sensing understanding) (Kampffmeyer, Salberg, & Jenssen, 2016; Long, Shelhamer, & Darrell, 2015). It involves labeling each pixel with the category of its enclosing object or region (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012; Farabet, Couprie, Najman, & LeCun, 2013; Hariharan, Arbeláez, Girshick, & Malik, 2014; Ning et al., 2005; Pinheiro & Collobert, 2014).

It is well known that an image consists of many pixels while the task of semantic segmentation comprises grouping or segmenting those pixels according to their semantic meaning. In the planning and development process of a city, diverse functional areas are formed. Like street view recognition and remote sensing understanding, semantics exists when cities are divided into grids. In a certain sense, the traffic states of cells in the grid (e.g., whether it is congested) can also be regarded as semantic information. Therefore, it is feasible to introduce techniques in semantic segmentation to enhance the accuracy of large-scale traffic prediction.

3 | PROBLEM DESCRIPTION

Consider a dense urban area with network-wide distributed e-hailing vehicle data. The study area can be partitioned into a grid, each element of which is called a cell. The workflow of an online car-hailing platform is to first perform cell-level prediction and then to dispatch the fleet. When a passenger makes a request, the dispatching system will match the available vehicle in the cell to the passenger. It should be noted that this topic is a business-driven practical problem, where

the cell unit is not universally feasible for different base models. The objective of this paper is to predict the traffic state for each cell in the grid in the following hour and effectively combine the output of multiple models. The intensity of demand is defined as the number of orders placed during a fixed interval within a cell. The flow is defined as the number of vehicles observed during a fixed interval within a cell.

We use $s_t^{i,j}$, $x_t^{i,j}$, and $d_t^{i,j}$ to denote the speed, flow, and demand intensity, respectively, in the t-th time interval at cell (i, j). The problem addressed in this paper is to use the historical data to predict $s_t^{i,j}$, $x_t^{i,j}$, and $d_t^{i,j}$ at each cell, which is a typical time series prediction problem.

The emphasis in time-series analysis of traffic prediction is usually onefold, for example, predict using only cell-level and macroscopic information. However, this limits the full utilization of the available data. As global information can indicate the overall trend of the time series while local fluctuation can reflect changes in each impact factor, it is imperative to consider both global and local information while conducting prediction.

For clear understanding, the various terms used in this paper are defined as follows.

Definition 1. Cell-level, local, and global information.

Because the study area has been partitioned into a grid of cells, the traffic state of a single cell is called cell-level information. Cell-level information is susceptible to the traffic states of geographically contiguous cells. Therefore, the states of a cell's neighbors (e.g., a 3×3 area) should be considered collectively, which is called local information. The aggregated traffic state data of all the cells are referred to as global information. The global or local information, indicating the variations in time series, is often neglected in the previous studies.

The notations used in this paper are listed below.

Notations:

S: Cell-level speed time series.
X: Cell-level flow time series.
D: Cell-level demand intensity time series.
L: Global speed series.
C: Global flow series.
O: Global demand intensity series.

Cell-level information can be expressed as

$$S = \left\{ s_{t-1}^{i,j},\ s_{t-2}^{i,j},\ s_{t-3}^{i,j}, \dots \right\} \tag{1}$$

$$X = \left\{ x_{t-1}^{i,j},\ x_{t-2}^{i,j},\ x_{t-3}^{i,j}, \dots \right\} \tag{2}$$

$$D = \left\{ d_{t-1}^{i,j},\ d_{t-2}^{i,j},\ d_{t-3}^{i,j}, \dots \right\} \tag{3}$$

The terms used for the global information are defined as follows:

$$L = \left\{ l_{t-1},\ l_{t-2},\ l_{t-3}, \dots \right\} \tag{4}$$

$$l_t = \frac{1}{p \cdot q} \sum\nolimits_1^p \sum\nolimits_1^q s_t^{i,j} \tag{5}$$

where (i, j) denote the index of the cell, p and q denote the maximum index values, and l_t denotes the average flow of all cells in the t-th time interval.

$$C = \left\{ c_{t-1},\ c_{t-2},\ c_{t-3}, \dots \right\} \tag{6}$$

$$c_t = \frac{1}{p \cdot q} \sum\nolimits_1^p \sum\nolimits_1^q x_t^{i,j} \tag{7}$$

where (i, j) denote the index of the cell, p and q denote the maximum index values, and c_t denotes the average speed of all cells in the t-th time interval.

$$O = \left\{ o_{t-1},\ o_{t-2},\ o_{t-3}, \dots \right\} \tag{8}$$

$$o_t = \frac{1}{p \cdot q} \sum\nolimits_1^p \sum\nolimits_1^q d_t^{i,j} \tag{9}$$

where (i, j) denote the index of the cell, p and q denote the maximum index values, and o_t denotes the average demand of all cells in the t-th time interval.

To address the limitations of the existing ensemble methods to handle spatio-temporal data, we propose a spatio-temporal ensemble method to incorporate this information.

In Figure 1, a modular design, constituting a traffic state prediction module and an ensemble module, is demonstrated. The first module contains multiple single-base learners for traffic state forecast while the model transformation is abstracted as several blocks in the figure. For example, the block "Base model 1" in Figure 1 represents the first traffic state prediction base model. Here, the base models can be any predictive model, including those based on machine learning methods and traffic flow theory. The output of a model is a vector, depicted as a long block. Because the city has been divided into a grid, this vector can be mapped onto the grid based on the indices. The vector is mapped to several k-by-k grids while the result of each model is treated as a channel of the image. Multiple channels are aggregated and used as the input of the ensemble module. The ensemble module is a fully convolutional model based on semantic segmentation technology. The feature map in Figure 1, a key concept in deep learning, represents the output of a layer in the neural network. The changes in the dimension of the feature map can be more directly observed from a three-dimensional perspective. The details of the framework are discussed in Section 6.

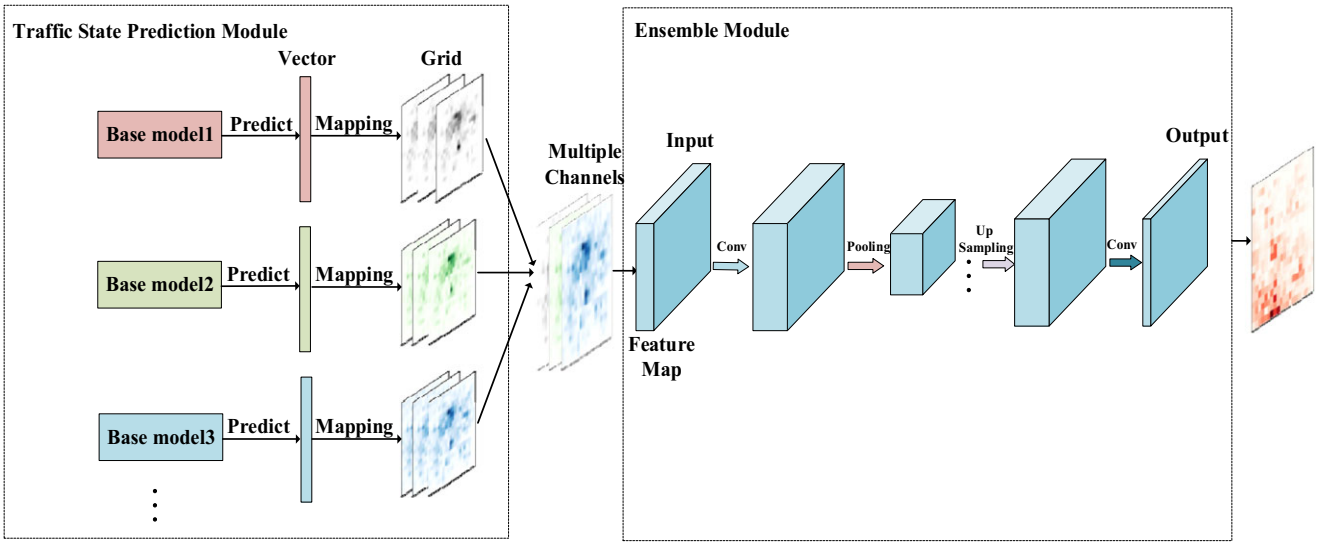


FIGURE 1 Framework overview of spatio-temporal ensemble net

4 | CONVOLUTIONAL NEURAL NETWORKS

In deep learning, convolutional neural network is a class of deep, feedforward artificial neural networks that employ at least one convolutional layer (Krizhevsky et al., 2012). The idea of convolutional networks stems from the resemblance between the connectivity among neurons and the biological organization of the animal visual cortex (Matsugu, Mori, Mitari, & Kaneda, 2003). It is suitable for processing data with grid-like structure, for example, images and audios that can be converted to two-dimensional and one-dimensional grids, respectively. The key advantage of using the convolutional layer over a fully connected layer is that the former enables sparse connectivity and parameter sharing.

4.1 | Convolution operation

Convolution operates on two functions with real values, denoted by f and k, which is formulated as follows:

$$\begin{aligned} h(x) &= (f^* k)(x) \\ &= \int f(\tau)k(x-\tau)d\tau \end{aligned} \tag{10}$$

The argument f in Equation (10) is often referred to as the input, and the function k as the kernel.

For discrete functions, the convolution operation is formulated as follows:

$$\begin{aligned} h(x) &= (f^* k)(x) \\ &= \sum_{\tau=-\infty}^{\infty} f(\tau)k(x-\tau) \end{aligned} \tag{11}$$

In the application of machine learning, the input is a multi-dimensional array of data, and the kernel is a multi-dimensional array of parameters that is learned by the learning algorithm. Provided both the input F and the kernel K are two-dimensional, then

$$\begin{aligned} H(i,j) &= (F^* K)(i,j) \\ &= \sum_m \sum_n F(m,\ n)K(i-m, j-n) \end{aligned} \tag{12}$$

As convolution operation is commutative, it can be rewritten as follows:

$$\begin{aligned} H(i,j) &= (F^* K)(i,j) \\ &= \sum_m \sum_n F(i-m, j-n)K(m,\ n) \end{aligned} \tag{13}$$

Note that many neural network libraries apply a cross-correlation function instead of the real convolution function for more convenient implementation. The formulation of the cross-correlation function is similar to the convolution function

$$\begin{aligned} H(i,j) &= (F^* K)(i,j) \\ &= \sum_m \sum_n F(i+m, j+n)K(m,\ n) \end{aligned} \tag{14}$$

4.2 | Sparse connectivity

A high-resolution image can introduce numerous neurons in a network. For instance, given an image of 1,000 × 1,000 pixels, there will be 1 million neurons in the input layer and another 1 million neurons in the hidden layer. For a fully connected neural network, as demonstrated in Figure 2, every input neuron is connected to a neuron in the subsequent hidden layer. Thus, there will be 10^6 parameters for every neuron in the hidden layer, resulting in an extensive network with a total of $10^6 \times 10^6$ training parameters. However, it is hardly practical to train such a large network due to both computational capabilities and cost limitations.

To overcome the drawback of involving an inordinate number of parameters in the fully connected neural net, locally

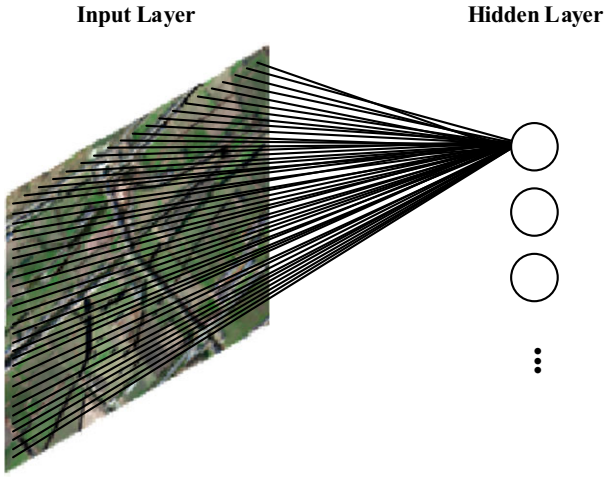


FIGURE 2 A fully connected neural net

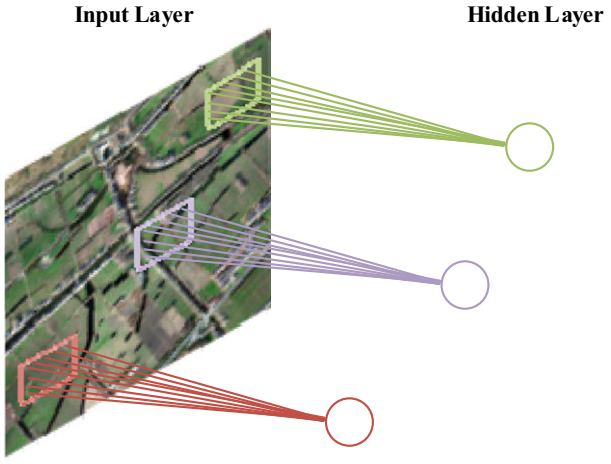


FIGURE 3 A locally connected neural net

connected method is adopted in convolutional neural networks. As demonstrated in Figure 3, in the case of a locally connected neural net, the neurons in the hidden layer are merely connected at a local region, for example, a 10×10 grid. In this way, the total number of parameters can be reduced to $10^6 \times 10^2$, which is much smaller than that of a fully connected neural network. As the size of the local area is much smaller than that of input, the finally obtained network is sparsely connected

As shown in Figure 4, although the number of connections is limited in a deep convolutional network, neurons in the deeper layer may still indirectly interact with a wide range of cells in the input layer. In this way, the network learns complex interactions between different variables efficiently by establishing the sparse connectivity.

In Figure 4, h1 can be directly influenced by three neurons, that is, x1, x2, and x3, which are called the receptive field of h1. Likewise, it can be observed that the receptive field of h4 contains x1, x2, x3, x4, and x5. It can be observed that

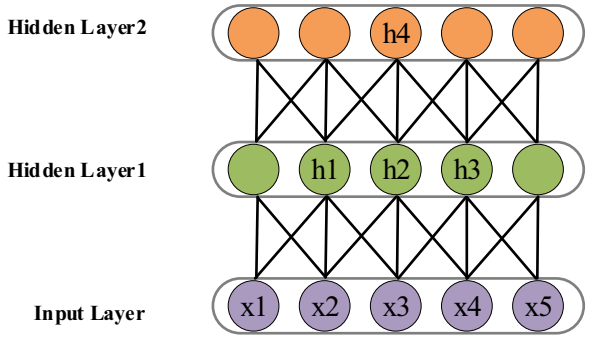


FIGURE 4 Sparse connectivity and receptive field

the receptive fields of neurons in deeper layers are larger than those in the shallow layers, indicating that, though connections of neurons in a convolutional neural network are sparse, it is still possible for the neurons in deeper layers to be indirectly connected to a large range of the input neurons.

4.3 | Parameter sharing

Parameter sharing is motivated by the observation that, in computer vision tasks, a filter used in one part of the image is still useful in another part of the image as well. For instance, Sobel filter (Sobel, 1978), a popular filter for edge detection, can be used to highlight edges in the whole image.

In the Sobel filter, two 3×3 kernels, each of which is the transpose of the other, are used to convolve with the input image and calculate the approximated derivatives for horizontal and vertical orientation. Let us denote $\mathbf{A}$ as the input image. $\mathbf{G}_x$ and $\mathbf{G}_y$ are the two images storing the approximation results for the two orientations, respectively, where $\mathbf{G}_x$ detects the horizontal edges and $\mathbf{G}_y$ detects the vertical edges. The computation process is shown as follows:

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \tag{15}$$

$$\mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A} \tag{16}$$

where the operator $*$ denotes the two-dimensional convolution operation.

For each pixel of the input image, the magnitude component of edge detection result can be expressed as

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2} \tag{17}$$

It can be found that the Sobel filter is connected to the input image merely through a kernel of size 3×3 and a fixed number of parameters, that is, 18. However, it can be applied to the whole image.

The main distinction between parameter sharing and Sobel filter is that in parameter sharing, the parameters are learned by the network rather than fixed values. Instead of learning a separate series of parameters for every 10×10 grid, with parameter sharing, we only need to learn one of the series and can apply it to the whole input image. In other words, the same convolutional kernel can be used for the convolution operation to be performed on the whole image. In practice, a convolution layer extracts a feature by sliding a fixed-sized filter over the image at a certain step length. Further, according to the requirement, more filters can be added to extract different types of features. Suppose there are 1,000 filters, the total number of parameters will be $10^3 \times 10^2$. It should be noted that a large amount of training data is required to ensure better performance with parameter sharing.

5 | DATA ANALYSIS

5.1 | Data description

The data used in this paper are extracted from the online car-hailing records in a part of Chengdu, China, which contains a large number of trip trajectory (more than 40 million trajectory records per day) and order data (more than 7 million order records in total) in November, 2016. GPS trajectory records contain anonymized driver id, user id, timestamp, and the corresponding latitude and longitude. Each trip order record comprises the information of anonymous order id, starting time, ending time, starting position, and ending position. The longitudinal and latitudinal range of the study area is 104.125E-104.045E and 30.725N-30.655N, respectively, covering approximately 59.51 km^2. The study area is partitioned into a 20×20 grid according to the range covered by the GPS trajectory data. The grid density in this study is set according to the real business demand of Didi. Note that the proposed methodology is applicable to other cities as well. Within a fixed interval (e.g., 1 hr), the traffic state (i.e., speed, flow, and demand intensity) in each cell could be easily aggregated based on the GPS trajectory/trip order data. The spatial distribution of the trajectory data is demonstrated in Figure 5. The kernel density estimate of the starting latitude and longitude of the demand is shown in Figure 6.

The stability and robustness of the data-collecting system are of great significance in the prediction task. Data sources like RFID and loop detectors are vulnerable to equipment failure and malfunctions. However, as the systems used in online car-hailing platforms are designed to minimize error, this ensures the transaction safety, service stability, as well as high data quality and reliability. For example, the client-side requests can be re-uploaded in case of a server failure. With the significant advances in communication technology,

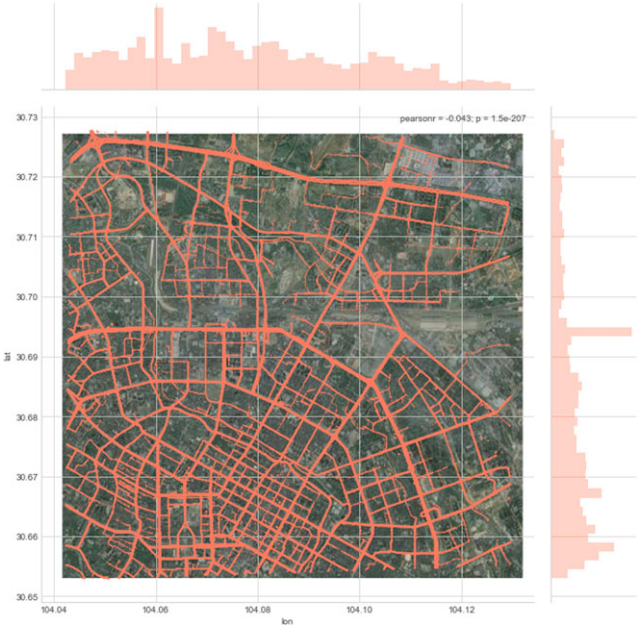


FIGURE 5 The latitude and longitude of the trajectory data

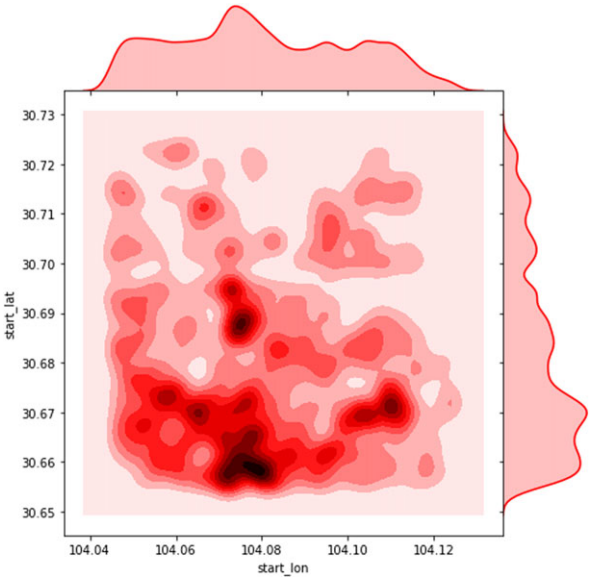


FIGURE 6 Kernel density estimate of the latitude and longitude of the demand

current network protocols address the issue of data package losses well. Moreover, in this paper, the data are aggregated based on a grid, whereby the influence of abnormal data can be further reduced.

5.2 | The characteristics of traffic state

To make the most of the information from multifarious scopes, many factors that used to be ignored need to be taken

into account. It is quite challenging to model these complicated factors simultaneously.

5.2.1 | Time smoothness

Time series data are intrinsically continuous rather than discrete. Hence, time series seldom produce sharp variations while the traffic state at a certain time interval has a close resemblance to traffic state at the adjacent time interval.

5.2.2 | Trend

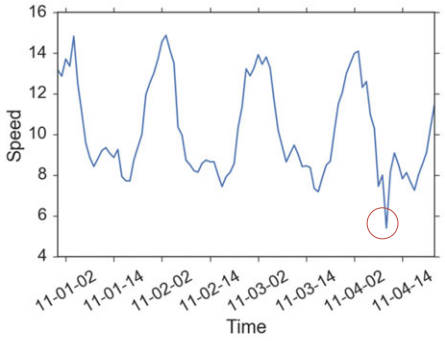
Generally, the traffic state of car-hailing service follows a particular trend, either increasing or decreasing. Such a trend implies how the traffic state will develop in the future.

5.2.3 | Period

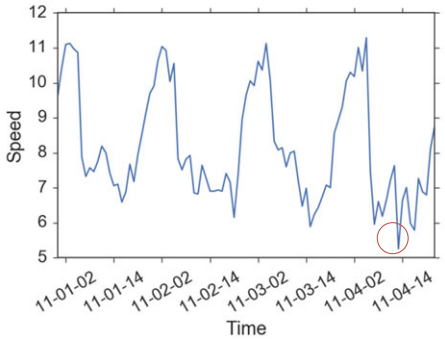
The similarity between the traffic state on two different days can be attributed to the periodic characteristics of traffic states, which tend to repeat every 24 hr (Jiang & Adeli, 2004). However, such periodic features do not necessarily mean the traffic pattern is immutable. Only when the travelers are dominated by commuting passengers, the traffic data can reflect the relatively fixed behavior patterns of passengers reasonably well. For example, the variations in passenger flow during a certain period are similar to that in the previous day. This factor can be particularly helpful in improving prediction. Likewise, the supply and demand for taxis in urban areas and the metro passenger flow also share similar characteristics.

5.2.4 | Spatial autocorrelation

A city is usually partitioned into different zones by planners according to various functions and attributes, for example, residential zones, commercial zones, and industrial zones. The traffic state of adjacent zones that share similar functions might be alike due to the spatial autocorrelation. Additionally, when traffic congestion occurs in certain regions of the urban road network, it tends to propagate upstream, thus causing traffic congestion nearby. An example of the congestion propagation can be observed in Figure 7, where the congestion diffuses from the cell (1, 1) to the cell (2, 1). The minimum point in Figure 7a indicates that congestion occurs at that time in a cell (1, 1). Apparently, the minimum point in Figure 7b appeared a short time after the one in Figure 7a, indicating that the congestion had diffused from cell (1, 1) to cell (2, 1). Because we have divided the network into a grid, such spatial characteristics can be better captured by Convolution Neural Networks in the ensemble module, which is introduced in the following Section 6.3. The hourly trip demand intensity is depicted in Figure 8, from which the demand intensity of the grids can be identified.



(a) Hourly Speed in Cell (1, 1).



(b) Hourly Speed in Cell (2, 1).

FIGURE 7 The process of traffic congestion propagation. (a) Hourly speed in cell (1, 1). (b) Hourly speed in cell (2, 1)

6 | THE SPATIO-TEMPORAL ENSEMBLE METHOD

6.1 | Framework overview

This method consists of two modules. The first module is termed as traffic state prediction module, which consists of multiple single-base learners for traffic state forecast. The second one, which is used for combining the output of base learners, is termed as ensemble module. Note that in the traffic state prediction module, different models and multi-source data like weather can be combined to improve prediction result. The emphasis of this study is to put forward a universal framework that can achieve a higher prediction accuracy through the ensemble of the prediction results of multiple base models, and there is no limit on the methods and number of them.

Regarding our method, we include temporal features in the base models. The base models predict the time series at one cell. Temporal features are necessary for base models to learn from historical data so as to generate the prediction results. However, it is difficult to incorporate the impacts from neighboring cells when designing features for a specific cell in the base models. In other words, the spatial features are hard to be directly included in base models. Therefore, the convolutional neural network is adopted in the ensemble module. The convolution operation is capable of capturing spatial features but is not suitable for modeling temporal features. The

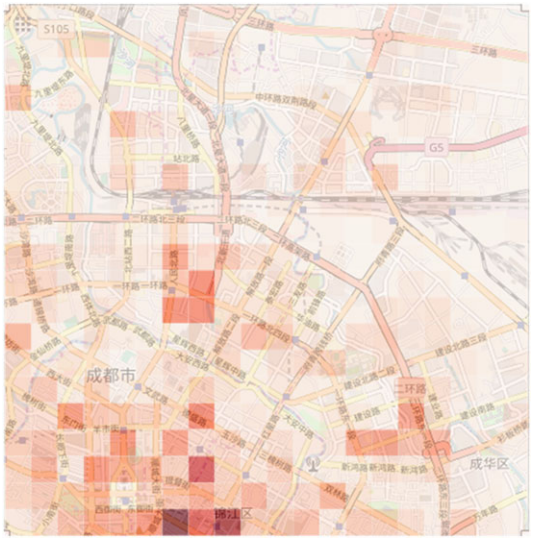


FIGURE 8 The demand intensity for partitioned grid

ensemble module and prediction module can compensate for each other to handle spatio-temporal information.

6.2 | Traffic state prediction module

Clearly, the flows at the current time interval share a higher similarity with those in the recent time interval than with the more distant ones. Existing time series prediction method often uses merely the information in the recent time interval to make the prediction. Apart from it, we further consider information from multi-view in our study.

Various features, including both the basic information features and the time series features, are designed and incorporated in the prediction models. The basic features contain the essential information like the cell index, the day of week, and hour. Macroscopic time series features contain median speed time series C, flow time series L, and demand time series O. Microscopic time series features include median speed time series X, flow time series S, and demand time series D. Time smoothness features, periodic features, trend features, and statistical features have been included in the original traffic state time series as it can be easily constructed. The time smoothness features are the information in adjacent time intervals, for example, $x_{t-1}^{i,j}$, $x_{t-2}^{i,j}$, and $x_{t-3}^{i,j}$. Periodic features denote the traffic state at the same time interval on the previous day, that is, $x_{t-24}^{i,j}$. Trend features comprise first-order trend and second-order trend. The first-order trend denotes the difference between traffic state in any two contiguous time periods, whereas the second-order trend denotes the difference between any two contiguous first-order traffic state trends, reflecting the changes in trend. Statistical features are statistics like variance and mean.

With the aim to achieve a higher prediction accuracy through the ensemble of different base models, a universal framework is proposed in the study that allows flexibility in selecting different models as per the data availability and study purpose. A similar method of partitioning data into subsets as stacking is applied to each model to predict all data (see Figure 9). The dataset is split into three folds, two of which are taken as training set and the remaining is used for prediction. In this way, the prediction results of the entire dataset can be obtained, as presented in Figure 9 (with Model 1 as an example). Finally, the prediction results are reshaped to a fixed-sized grid. Because the city has been divided into a grid, this vector can be mapped onto the grid based on the indices (see Figure 10).

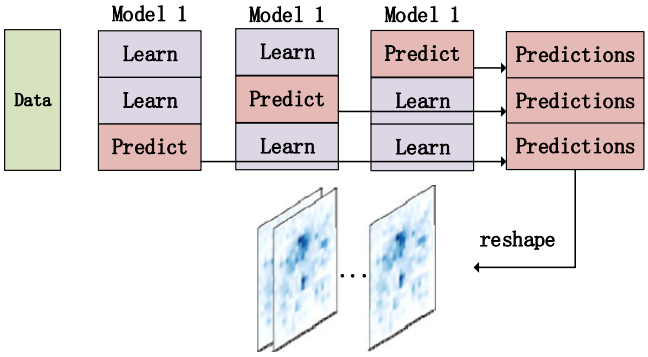


FIGURE 9 Traffic state prediction module

Grid index (i,j)	Predicted value
(0,0)	6
(0,1)	7
(0,2)	9
…	…
(k,k)	15

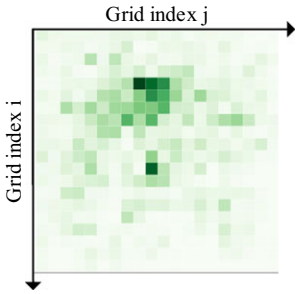


FIGURE 10 Mapping process

6.3 | Ensemble module

In Section 5.2, the spatial autocorrelation phenomenon is briefly introduced. Traffic congestion that occurs in a particular cell is likely to propagate to cells nearby, thereby causing traffic congestion in such cells. In other words, the traffic state of a cell is susceptible to that of geographically contiguous cells.

It would be a complicated task to investigate this problem using traditional methods as it might involve a large amount of computation. For any randomly selected cell (i, j), multiple contiguous cells will have a combined impact on it. Thus, the complex superposed influence, rather than the independent influence of each surrounding cell, should be taken into consideration. For example, consider a 3×3 area surrounding

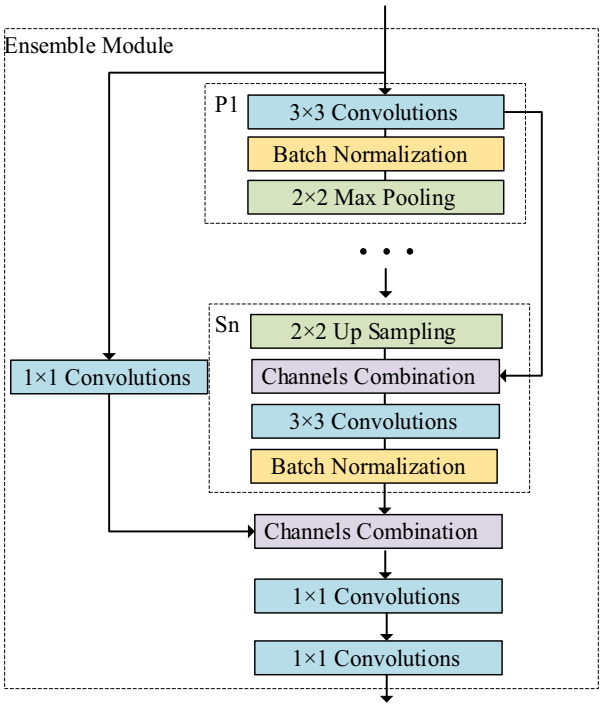


FIGURE 11 Ensemble module

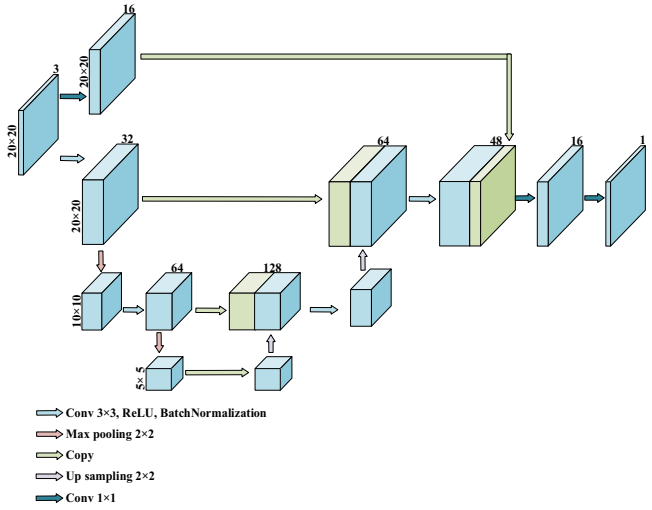


FIGURE 12 Dimensional changes in feature maps

a cell (including the cell itself). As for the influence that the traffic state of a single cell has on prediction (i.e., cell-level features), we can take the traffic states of the nine cells as features, with a set of weights to be learned by the model. For the influence of multiple cells' traffic states (i.e., local features), the features can be obtained by calculating predefined statistical indicators of several selected cells. Then, the 3×3 area can be expanded to 6×6, with the same procedure performed multiple times. Whether the result can meet the requirement depends on how the statistical indicators, that is, features, are designed, which relies significantly on previous experience and costs a great deal of time.

Convolution and pooling have provided a highly efficient solution to such a problem. With the help of a 3×3 convolutional kernel, the complicated influence on the cell (i, j) by the 3×3 surrounding area can be automatically extracted. Different features can also be extracted by using multiple convolutional kernels. Moreover, with 2×2 pooling, the receptive field can be further expanded, allowing for the influence of a larger area, for example, a 6×6 area on the cell (i, j).

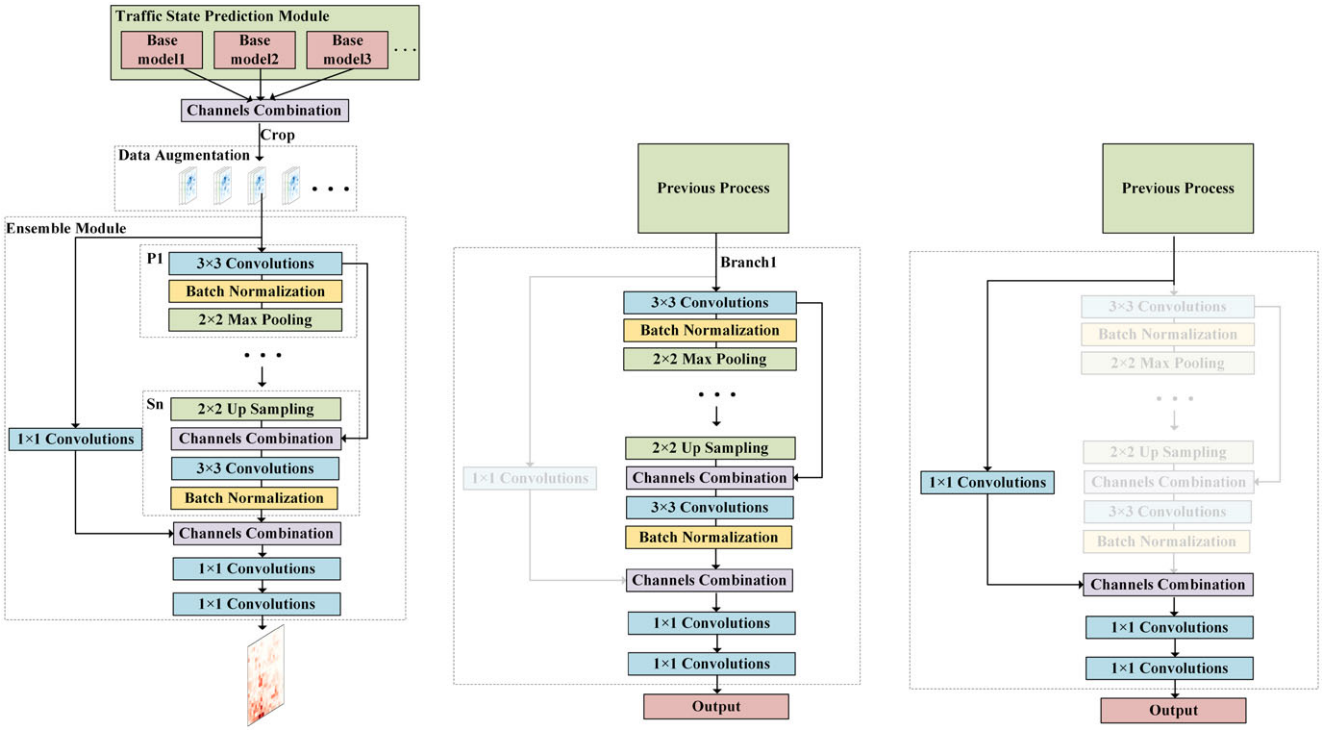
We aim to establish a "fully convolutional" network that can account for the inputs of an arbitrary scale and generate the output of homologous scales with efficient inference and learning. The ensemble module is demonstrated in Figure 11. The output of the three base traffic prediction models is reshaped to several k-by-k grids while the result of each model is treated as a channel of the image. The three channels together are used as the inputs of the ensemble module. Due to multiple pursuant processes of convolutions, batch normalization (Ioffe & Szegedy, 2015) and max-pooling, namely, (P1, P2, …, Pn), the model can learn features that are highly robust. Here, batch normalization is used to prevent overfitting and to reduce the sensitivity of the network to initial weights (Ioffe & Szegedy, 2015). After the pooling operation, the size of the feature map has shrunk (e.g., consider an input size of 20×20, the size of the feature map will shrink from 20×20 to 5×5 after performing two 2×2 pooling operations). As the prediction is made based on each pixel, it is required to restore the size of the feature map to 20×20.

The restoring process of the feature map is not achieved by merely applying upsampling once. Instead, multiple pursuant processes of upsampling, channels combination, convolutions, and batch normalization, namely, (S1, S2, …, Sn), are applied.

Notably, each of the three base models alone is able to present an acceptable accuracy. By introducing a 1×1 convolution, the dimension of the input is increased and combined with the previous output, thus ensuring that the overall accuracy of the proposed method is better than any of the original models. Finally, with the help of 1×1 convolution, the output is transformed to a single channel with its size the same as the input grid. It can be trained via backpropagation and Adam optimizing algorithm (Kingma & Ba, 2015).

Suppose our input is a three-dimensional 20×20 image. The change in the dimensionality of the feature map can be observed in Figure 12. Each of the blue rectangles denotes a multi-channel feature map. On the top of the rectangle, the number of channels is given with its size labeled on the left side. In addition, the gray rectangles correspond to duplicated feature maps, and the arrows represent various operations.

Our grid-based prediction is reasonable to some extent in terms of the feasibility in engineering implementation, and many large-scale traffic predictions in real applications are based on grid. Grid-based partitioning method is able to



(a) Spatio-temporal Ensemble Net (b) Branch 1 (c) Branch 2

FIGURE 13 Spatio-temporal ensemble net. (a) Spatio-temporal ensemble net. (b) Branch 1. (c) Branch 2

represent the prediction results of base models in the form of the image channels, which makes it easier to apply existing deep learning technologies, like convolutional neural network, and data augmentation. Conducting precise pixel-level prediction is an extremely challenging task, which needs many technologies relating to semantic segmentation. The feature map has to be compressed first to assure the model of learning the spatial pattern of traffic state, and the information is then reconstructed step-by-step. Skip connection makes it possible for the model to combine the information learnt in different layers of the network. The input of traditional machine learning methods is the extensive manually designed features. Instead, the proposed method goes beyond the conventional way and considers channels as its input, where each channel represents the result of a base model. The core concept of the method is channel combination, which means that the channels representing different base models can be combined not only in the input, but also in the network, different layers, and different branches. In this way, the information they learnt in different layers and branches can be gathered.

It should be noted that the weights of different base models are learned by the deep convolutional neural network in the ensemble module. Assuming there are three models, at least three coefficients are needed to weight the results. For different areas in the grid, using the same set of weights may not be reasonable. For example, in cell A, the weight for Model 1 is higher, whereas in cell B, the weight for Model 2 is higher. Provided that there are 400 cells, 400 × 3 coefficients are required. However, when setting weights for models in each cell, the impact from other cells cannot be considered. The multi-channel image-like input can precisely handle this issue. The convolutional neural network is therefore adopted to capture spatial features. Furthermore, the traditional approach requires one machine learning model to learn the weights of each cell, which means 400 extra models will be needed. Instead, the proposed method requires only a deep convolutional neural network to learn the weights, which can help reduce computational complexity. Moreover, the ensemble method based on deep convolutional neural network is capable of leveraging the computing power of GPU and is transferable to parallel deep learning systems.

7 | EXPERIMENTS

The structure of the spatio-temporal ensemble net is demonstrated in Figure 13. The outputs of the traffic state prediction module are transformed into 696 images. Then, data augmentation technique is used to increase the quantity of data. As shown in Figure 14, the size of the original grid is 20 × 20. By using a 10 × 10 sliding window with step length 1, 69,600 images can be obtained, with 70% of them (i.e., 48,720

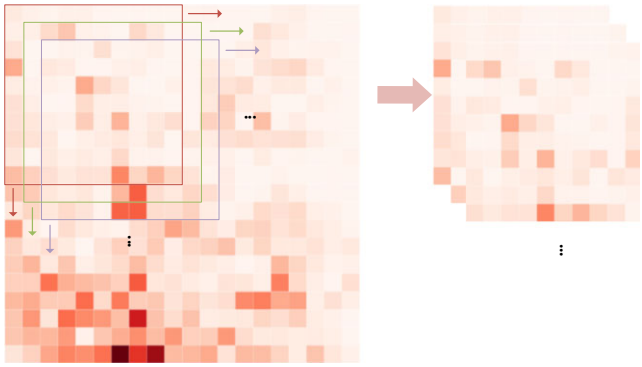


FIGURE 14 Data augmentation

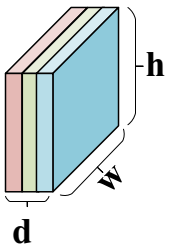


FIGURE 15 The input data of ensemble module

images) set aside for training the ensemble model. The input size of the ensemble module is 10×10.

The channels (i.e., outputs) of the base models in the traffic state prediction module are first combined and then cropped by data augmentation. The cropped images are fed into the ensemble module as the inputs. As shown in Figures 13b and 13c, there are two branches in the ensemble module. A branch is defined as a runnable subnetwork. In the experiment, we further test the two branches of the model for comparison. Branch 1 does not fuse the result that increases the dimension of the original input directly. Branch 2 uses 1×1 convolution filter to reduce the dimensionality.

The task of prediction on a pixel level is far more difficult than image classification. The proposed spatio-temporal ensemble net is specifically designed in the following aspects:

1. First compress the feature map, and then restore. The input data of the ensemble module, as shown in Figure 15, are an image of size h, w, and d, where h and w are spatial dimensions (i.e., the height and width of the image) and d is the channel dimension. Here, the channels represent the prediction results of base models (a standard image contains three channels, i.e., red, green, and blue). It should be noted that, in this study, the prediction result of each base model is treated as one channel. Observing the dimensional changes in Figure 16, the spatial dimensions are first compressed and then restored.

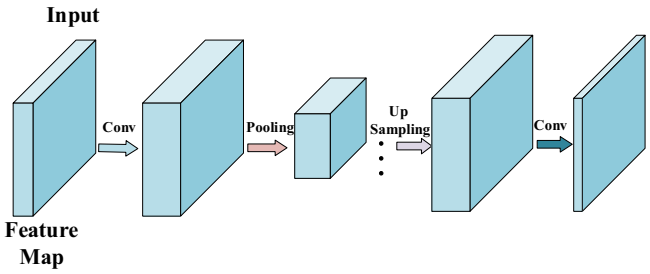


FIGURE 16 Dimensional changes in feature maps

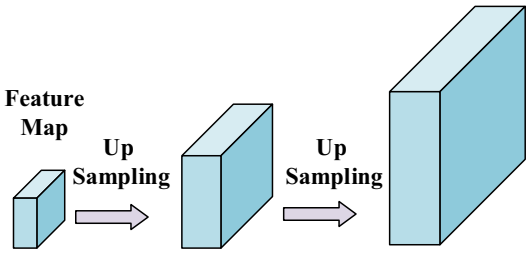


FIGURE 17 Restoring process of feature maps

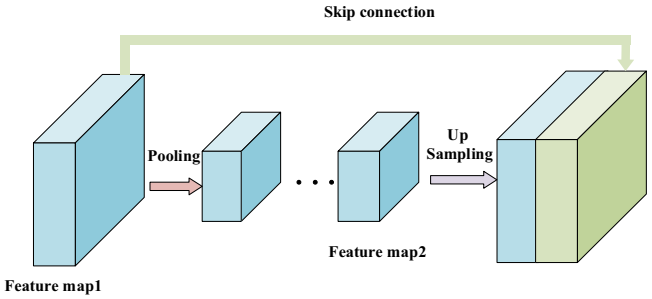


FIGURE 18 Skip connection

2. The restoring process of the feature map is not achieved by merely applying upsampling once but by restoring step-by-step (see Figure 17).
3. Deep coarse information and shallow fine information are connected using a kind of "skip" architecture. As shown in Figure 18, feature map 1 is the output of a shallow hidden layer, which contains fine-grained information, whereas feature map 2 is the output of a deep hidden layer, which contains coarse information obtained through the compression of pooling operation. Subsequently, an upsampling operation is performed on feature map 2 to obtain a new feature map, the size of which is the same as the feature map 1. Finally, the new feature map and feature map 1 are concatenated together by a skip connection.

We measure the performance of our method through mean absolute error (MAE), which is defined as

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{18}$$

TABLE 1 Performance comparison of MAE in the validation set

Model	Speed (km/h)	Flow (vph)	Demand density (oph)
Base model 1 (LightGBM)	0.914	31.998	3.832
Base model 2 (LightGBM)	0.957	30.232	4.054
Base model 3 (LightGBM)	0.906	31.202	3.919
Average (base models 1–3)	0.926	31.144	3.935
Branch 1	0.838	29.252	3.721
Branch 2	0.841	29.653	3.743
Spatio-temporal ensemble net	0.839	27.225	3.709

TABLE 2 Performance comparison of MAE in the validation set

Model	Speed (km/h)	Flow (vph)	Demand density (oph)
Base model 1 (LightGBM)	0.914	31.998	3.832
Base model 4 (KNN)	0.987	35.625	4.504
Base model 5 (linear regression)	1.014	35.409	4.423
Average (base models 1, 4, and 5)	0.971	34.344	4.253
Branch 1	0.859	29.621	3.736
Branch 2	0.871	29.814	3.757
Spatio-temporal ensemble net	0.850	27.654	3.721

where y_i is the i-th observed traffic state and $\hat{y}_i$ is the corresponding forecasting traffic state. N is the total number of forecasting traffic state samples.

The MAEs of different models in the validation set are listed in Tables 1 and 2. In Tables 1 and 2, the unit of speed, flow, and demand density is kilometers per hour (km/h), vehicles per hour (vph), and orders per hour (oph), respectively. The columns show the MAE of the models for each parameter. The MAEs in Tables 1 and 2 are obtained using Equation 18.

For code implementation, some sensible designs are utilized. The *EarlyStopping* rule is set to stop training when the performance has ceased to improve. Furthermore, to observe the internal status of the network, call-back functions, a set of functions to be applied at given stages of the training procedure, are defined. Then, statistics pertaining to the validation set can be displayed at the end of each epoch through this call-back function. In the experiment, the batch size is set to 128, whereas the initial learning rate of Adam algorithm is set to 0.001.

LightGBM, a light gradient boosting function library based on decision tree algorithm (Ke et al., 2017), is adopted in the traffic state prediction module. We used different feature sets, for example, macroscopic information set and microscopic information set, to generate multiple base models, with the purpose of simulating diverse models used in real scenarios. Base model 1 (LightGBM) uses all the features in the previous 24 hr. Base model 2 (LightGBM) uses the global features in the previous 24 hr. Base model 3 (LightGBM) uses the cell-level features in the previous 24 hr. In addition, some "bad" base models are also deliberately added. Base model 5 (linear regression) and base model 4 (KNN) are added as base models with bad performance. In Table 2, we use the same features for the three base models. From the experiment results listed in Table 1, it can be found that branch 1 and branch 2 are both able to effectively reduce the MAE of the base model regarding the prediction of speed, flow, and demand intensity. More precisely, the MAEs of speed predictions are found to decline from an average of 0.926 to 0.838 and 0.841. In the case of traffic flow, the MAEs have dropped from 31.144 to 29.252 and 29.653 while the demand density decline from 3.935 to 3.721 and 3.743. As shown in Figures 15b and 15c, branch 1 mainly depends on the 3×3 kernel to extract features, whereas branch 2 depends mostly on the 1×1 kernel. Combining kernels of various sizes can increase the adaptability of the network to scale. For this reason, in the spatio-temporal ensemble net, the feature maps are concatenated (i.e., channels combination), and the convolutional kernel in the next layer can automatically determine which channels are more accurate. By combining branches 1 and 2, the MAE of flow further drops to 27.225 while demand density decreases to 3.709.

In Table 2, we employ three base models with heterogeneous performance, whose MAE significantly differs from each other. The MAEs of the speed predictions are found to decline from an average of 0.971 to 0.850. In the case of traffic flow, the MAEs have dropped from 34.344 to 27.654, while that of the demand density decline from 4.253 to 3.721. Although some of the base models show poor prediction results, the MAEs can still be reduced by our proposed spatio-temporal ensemble net.

It should be noted that the performance of the ensemble net did not exceed the base models by a significant margin. However, in the context of the scenario in this paper, even a little improvement in accuracy is valuable, as considerable cumulative benefits will be brought to the online car-hailing service platform. For instance, in the demand prediction of car-hailing service, even if the accuracy only improves slightly, the cumulative results will be considerable. Without loss of generality, let demand = orders / (duration × area). Considering that the entire city can be divided into thousands of subareas, and there are hundreds of cities where online car-hailing platform run their business, the total reduction in prediction error due to a slight improvement in accuracy can be substantial.

As shown in Figure 15a, in the spatio-temporal ensemble net, three types of channel combination are designed, including directly combining the channels (i.e., the prediction results of base models), combining the outputs of different layers using a skip connection, and fusing the outputs of branch 1 and branch 2. Traditional machine learning methods heavily rely on feature engineering to produce features, and the weights of these features are learned by the models based on the designed features and the corresponding training data. The proposed method jumps out of the traditional process as the inputs are now channels rather than features while the importance of these channels can be efficiently determined by the model, a key advantage associated with channel combination.

Two sharing mechanisms are utilized in this paper, that is, the innate parameter sharing of convolutional neural network (see Section 4.3) and the higher-level sharing. In the process of data augmentation, to increase data size, a sliding window with a fixed size is applied to scan the raw image with a specific step length. This procedure is similar to the operation of convolution layer, where the fixed-sized convolution kernel is used, moved at a certain step length. As is mentioned in Section 4.2, the parameter of a convolution kernel used in one part of the image can be reused in another part of the image as well. The whole model is equivalent to an abstract parameter with a higher level herein. Further, the model used in one data-augmented sample is probably useful in another sample as well. This is the reason why data augmentation can generate effective results. It is quite complicated to extract features of local regions through traditional approach of manual feature designing, which involves difficulties in determining the size of the local region, which cells are likely to exert impact, what features should be extracted, and so forth. Therefore, in spatio-temporal ensemble net and branch 1, convolutional neural networks are employed to enable the automatic extraction of features instead of manual feature designing. Furthermore, even though the size of the convolutional kernel used is only 3×3, its receptive fields in the deeper layers are much larger. The receptive fields become larger in deeper layers, thus enabling it to extract the influence of varied-sized regions on prediction results.

Intuitively, compared with grid-based models, other methods like traffic-zone-based models might be more reasonable. However, traffic-zone-based models have both advantages and disadvantages like the difficulty of data processing, the versatility of the models, and the implementation difficulty. The disadvantages are summarized as follows:

1. Because there is no grid-like structure in traffic zones, we cannot extract features by convolutional neural network. Instead, a large number of features need to be manually extracted when designing the algorithm.
2. Traffic zones are city specific. Given the model is applied to another city, it must be modified to suit the characteristics of that place.
3. Algorithm relating to geofencing is needed while estimating which traffic zone the data falls in. Considering the time complexity, if the data size is extremely large, it cannot be completed within an acceptable time even with the help of a large-scale distributed computing.

Grid-based traffic prediction can be performed with convolutional neural network while the construction of a grid does not require any complicated geofencing algorithm. Many large-scale traffic predictions in applications like Didi, Microsoft Research Asia, etc. are grid based (Yao et al., 2018; Zhang et al., 2017). Moreover, grid-based prediction is not only applied in the traffic domain, but also in the meteorological field like rainfall and fine-grained air quality predictions.

8 | CONCLUSIONS

This paper focuses on the construction of an ensemble framework explicitly designed for large-scale traffic state prediction based on spatio-temporal data, where a fully convolutional model leveraging the semantic segmentation technology is proposed to improve the prediction performance. The proposed spatio-temporal ensemble net combines multiple traffic state prediction models and is trained on an end-to-end and pixels-to-pixels basis. Considering the limited data size, we applied the data augmentation technique to extend the size of the training data. Two modules constitute the proposed framework, namely, the traffic state prediction module and the ensemble module. The data are mapped into cells to facilitate the use of the convolutional neural network, whereby the spatial information can be incorporated. Combined with the base models that usually focus on manipulating cell-level temporal information, the two modules can compensate for each other to handle spatio-temporal information. Experimental results on city-wide predictions of speed, flow, and demand intensity suggest that through spatio-temporal ensemble net,

multiple traffic state prediction models can be combined to improve the prediction accuracy. According to the validation results on base models with homogeneous performance, all of the three prediction tasks have witnessed a decline in MAE. Moreover, when the base models display heterogeneous performance, which means some of them may have weak predictive power, our framework can still generate an improved result. In the future, it is recommended to apply more innovative thoughts and frameworks from semantic segmentation and object detection to the field of transportation, including large-scale spatio-temporal prediction and detection of traffic congestion identification.

ACKNOWLEDGMENTS

This study is supported by the General Projects (No. 71771050) and Key Projects (No. 51638004) of the National Natural Science Foundation of China, the Fundamental Research Funds for the Central Universities (No. 2242018K41023/2242019K41012), and the Scientific Research Foundation of Graduate School of Southeast University (YBPY1927).

REFERENCES

Adeli, H., & Ghosh-Dastidar, S. (2004). Mesoscopic-wavelet freeway work zone flow and congestion feature extraction model. *Journal of Transportation Engineering*, *130*(1), 94–103.

Adeli, H., & Jiang, X. (2009). *Intelligent infrastructure—Neural networks, wavelets, and chaos theory for intelligent transportation systems and smart structures*. Boca Raton, FL: CRC Press, Taylor & Francis.

Adeli, H., & Karim, A. (2005). *Wavelets in intelligent transportation systems*. New York, NY: John Wiley & Sons.

Ahmed, M. S., & Cook, A. R. (1979). Analysis of freeway traffic time-series data by using Box–Jenkins techniques. *Transport Research Record*, *722*, 1–9.

Allström, A., Ekström, J., Gundlegård, D., Ringdahl, R., Rydergren, C., Bayen, A. M., & Patire, A. D. (2016). Hybrid approach for short-term traffic state and travel time prediction on highways. *Transportation Research Record: Journal of the Transportation Research Board*, *2554*, 60–68.

Antoniou, C., Koutsopoulos, H. N., & Yannis, G. (2013). Dynamic data-driven local traffic state estimation and prediction. *Transportation Research Part C: Emerging Technologies*, *34*, 89–107.

Bertini, R. L., & Leal, M. T. (2005). Empirical study of traffic features at a freeway lane drop. *Journal of Transportation Engineering*, *131*(6), 397–407.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks: Deep learning-based crack damage detection using CNNs. *Computer-Aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. *Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, Nevada*, 2843–2851.

Daganzo, C. (1997). *Fundamentals of transportation and traffic operations* (1st ed.). Oxford, NY: Pergamon.

Deng, D., Shahabi, C., Demiryurek, U., & Zhu, L. (2017). Situation aware multi-task learning for traffic prediction. IEEE International Conference on Data Mining (ICDM), IEEE, New Orleans, LA, 81–90.

Dharia, A., & Adeli, H. (2003). Neural network model for rapid forecasting of freeway link travel time. *Engineering Applications of Artificial Intelligence*, *16*(7-8), 607–613.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In G. Goos, J. Hartmanis, & J. van Leeuwen (Series Eds.), *Multiple classifier systems* (Vol. 1857, pp. 1–15). Berlin, Heidelberg: Springer.

Duan, Y., Lv, Y., Liu, Y.-L., & Wang, F.-Y. (2016). An efficient realization of deep learning for traffic data imputation. *Transportation Research Part C: Emerging Technologies*, *72*, 168–181.

Farabet, C., Couprie, C., Najman, L., & Lecun, Y. (2013). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1915–1929.

Fischer, P., Dosovitskiy, A., & Brox, T. (2014). Descriptor matching with convolutional neural networks: A comparison to SIFT. ArXiv:1405.5769. Retrieved from http://arxiv.org/abs/1405.5769

Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, Wuhan, China, 324–328.

Fusco, G., Colombaroni, C., Comelli, L., & Isaenko, N. (2015). Short-term traffic predictions on large urban traffic networks: Applications of network-based machine learning models and dynamic traffic assignment models. International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), IEEE, Budapest, Hungary, 93–101.

Ghosh, B., Basu, B., & O'Mahony, M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE Transactions on Intelligent Transportation Systems*, *10*(2), 246–254.

Gu, X., Abdel-Aty, M., Xiang, Q., Cai, Q., & Yuan, J. (2019). Utilizing UAV video data for in-depth analysis of drivers' crash risk at interchange merging areas. *Accident Analysis & Prevention*, *123*, 159–169.

Guo, Y., Liu, P., Wu, Y., & Chen, J. (2018). Evaluating how right-turn treatments affect right-turn-on-red conflicts at signalized intersections. *Journal of Transportation Safety & Security*. https://doi.org/10.1080/19439962.2018.1490368

Guzman, E., El-Haliby, M., & Bruegge, B. (2015). Ensemble methods for app review classification: An approach for software evolution (N). 30th IEEE/ACM International Conference on Automated Software Engineering (ASE), IEEE, Lincoln, NE, 771–776.

Han, K., Piccoli, B., & Friesz, T. L. (2016). Continuity of the path delay operator for dynamic network loading with spillback. *Transportation Research Part B: Methodological*, *92*, 211–233.

Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. (2014). Simultaneous detection and segmentation. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *The European conference on computer vision* (Vol. 8695, pp. 297–312). Cham, Switzerland: Springer.

Hashemi, H., & Abdelghany, K. F. (2016). Real-time traffic network state estimation and prediction with decision support capabilities: Application to integrated corridor management. *Transportation Research Part C: Emerging Technologies*, *73*, 128–146.

Hashemi, H., & Abdelghany, K. (2018). End-to-end deep learning methodology for real-time traffic network management: Deep learning for real-time traffic network management. *Computer-Aided Civil and Infrastructure Engineering*, *33*(10), 849–863.

He, K., Zhang, X., Ren, S., & Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *The European conference on computer vision* (Vol. 8691, pp. 346–361). Cham, Switzerland: Springer.

Hofleitner, A., Herring, R., Abbeel, P., & Bayen, A. (2012). Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1679–1693.

Huang, W., Song, G., Hong, H., & Xie, K. (2014). Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*, *15*(5), 2191–2201.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 37*, 448–456.

Jiang, X., & Adeli, H. (2004). Wavelet packet-autocorrelation function method for traffic flow pattern analysis. *Computer-Aided Civil and Infrastructure Engineering*, *19*(6), 324–337.

Jiang, X., & Adeli, H. (2005). Dynamic wavelet neural network model for traffic flow forecasting. *Journal of Transportation Engineering*, *131*(10), 771–779.

Kamarianakis, Y., & Prastacos, P. (2003). Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record: Journal of the Transportation Research Board*, *1857*, 74–84.

Kampffmeyer, M., Salberg, A.-B., & Jenssen, R. (2016). Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, IEEE, Las Vegas Valley, NV, 1–9.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., … Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.),*Advances in neural information processing systems* (Vol. 30, pp. 3146–3154). Long Beach, CA: Curran Associates.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations, San Diego, CA*, 1–15.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV*, 1097–1105.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Hoboken, NJ: John Wiley & Sons.

Li, L., Zhang, J., Wang, Y., & Ran, B. (2018). Missing value imputation for traffic-related time series data based on a multi-view learning method. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. https://ieeexplore.ieee.org/document/8478191

Lin, Y., Nie, Z., & Ma, H. (2017). Structural damage detection with automatic feature-extraction through deep learning: Structural damage detection with automatic feature-extraction through deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *32*(12), 1025–1046.

Liu, P., Wu, J., Zhou, H., Bao, J., & Yang, Z. (2019). Estimating queue length for contraflow left-turn lane design at signalized intersections. *Journal of Transportation Engineering, Part A: Systems*, *145*(6), 04019020.

Liu, Y., Jia, R., Xie, X., & Liu, Z. (2019). A two-stage destination prediction framework of shared bicycle based on geographical position recommendation. *IEEE Intelligent Transportation Systems Magazine*, *11*(1), 42–47.

Liu, Y., Liu, Z., & Jia, R. (2019). DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C: Emerging Technologies*, *101*, 18–34.

Liu, Z., Wang, S., Zhou, B., & Cheng, Q. (2017). Robust optimization of distance-based tolls in a network considering stochastic day to day dynamics. *Transportation Research Part C: Emerging Technologies*, *79*, 58–72.

Lo, H. K., & Szeto, W. Y. (2002). A cell-based dynamic traffic assignment model: Formulation and properties. *Mathematical and Computer Modelling*, *35*(7–8), 849–865.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE, Boston, MA, 3431–3440.

Long, J. L., Zhang, N., & Darrell, T. (2014). Do convnets learn correspondence? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 27, pp. 1601–1609). Montreal, Canada: Curran Associates.

Lu, C.-C., & Zhou, X. (2014). Short-term highway traffic state prediction using structural state space models. *Journal of Intelligent Transportation Systems*, *18*(3), 309–322.

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2014). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2), 865–873.

Ma, S., Zheng, Y., & Wolfson, O. (2013). T-share: A large-scale dynamic taxi ridesharing service. IEEE 29th International Conference on Data Engineering (ICDE), IEEE, Brisbane, Australia, 410–421.

Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, *54*, 187–197.

Ma, X., Yu, H., Wang, Y., & Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PLoS One*, *10*(3), e0119044.

Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, *16*(5-6), 555–559.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. 11th Annual Conference of the International Speech Communication Association. Makuhari, Chiba, Japan, 1045–1048.

Mo, B., Li, R., & Zhan, X. (2017). Speed profile estimation using license plate recognition data. *Transportation Research Part C: Emerging Technologies*, *82*, 358–378.

Nabian, M. A., & Meidani, H. (2018). Deep learning for accelerated seismic reliability analysis of transportation networks: Deep learning for

network reliability analysis. *Computer-Aided Civil and Infrastructure Engineering*, *33*(6), 443–458.

Nantes, A., Ngoduy, D., Bhaskar, A., Miska, M., & Chung, E. (2016). Real-time traffic state estimation in urban corridors from heterogeneous data. *Transportation Research Part C: Emerging Technologies*, *66*, 99–118.

Nanthawichit, C., Nakatsuji, T., & Suzuki, H. (2003). Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. *Transportation Research Record: Journal of the Transportation Research Board*, *1855*, 49–59.

Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., & Barbano, P. E. (2005). Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, *14*(9), 1360–1371.

Niu, X., Zhu, Y., & Zhang, X. (2014). DeepSense: A novel learning mechanism for traffic prediction with taxi GPS traces. IEEE Global Communications Conference, IEEE, Austin, TX, 2745–2750.

Oh, S., Byon, Y.-J., & Yeo, H. (2016). Improvement of search strategy with k-nearest neighbors approach for traffic state prediction. *IEEE Transactions on Intelligent Transportation Systems*, *17*(4), 1146–1156.

Pan, T. L., Sumalee, A., Zhong, R. X., & Indra-payoong, N. (2013). Short-term traffic state prediction based on temporal–spatial correlation. *IEEE Transactions on Intelligent Transportation Systems*, *14*(3), 1242–1254.

Pan, Y., Chen, S., Qiao, F., Ukkusuri, S. V., & Tang, K. (2019). Estimation of real-driving emissions for buses fueled with liquefied natural gas based on gradient boosted regression trees. *Science of the Total Environment*, *660*, 741–750.

Pinheiro, P. O., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. Proceedings of the 31st International Conference on Machine Learning, *JMLR, Beijing, China, 32*, 82–90.

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *79*, 1–17.

Qi, Y., & Ishak, S. (2014). A Hidden Markov Model for short term prediction of traffic conditions on freeways. *Transportation Research Part C: Emerging Technologies*, *43*, 95–111.

Rafiei, M. H., & Adeli, H. (2017). A novel machine learning-based algorithm to detect damage in high-rise building structures. *The Structural Design of Tall and Special Buildings*, *26*(18), e1400.

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Rafiei, M. H., Khushefati, W. H., Demirboga, R., & Adeli, H. (2017). Supervised deep restricted Boltzmann machine for estimation of concrete. *ACI Materials Journal*, *114*(2), 237–244.

Ramezani, M., & Geroliminis, N. (2012). On the estimation of arterial route travel time distribution with Markov chains. *Transortation Research Part B: Methodological*, *46*(10), 1576–1590.

Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1619–1630.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & Lecun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. *International Conference on Learning Representations (ICLR2014)*, Banff, Canada, 1–16.

Sheffi, Y. (1984). *Urban transportation networks: Equilibrium analysis with mathematical programming methods*. Englewood Cliffs, NJ: Prentice-Hall.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. ArXiv:1409.1556. Retrieved from http://arxiv.org/abs/1409.1556

Singh, S., Hoiem, D., & Forsyth, D. (2016). Swapout: Learning an ensemble of deep architectures. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 29, pp. 28–36). Barcelona, Spain: Curran Associates.

Smith, B. L., & Demetsky, M. J. (1994). Short-term traffic flow prediction models—A comparison of neural network and nonparametric regression approaches. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, *2*, 1706–1709.

Smith, B. L., & Demetsky, M. J. (1997). Traffic flow forecasting: Comparison of modeling approaches. *Journal of Transportation Engineering*, *123*(4), 261–266.

Sobel, I. (1978). Neighborhood coding of binary images for fast contour following and general binary array processing. *Computer Graphics and Image Processing*, *8*(1), 127–135.

Song, W., Han, K., Wang, Y., Friesz, T., & del Castillo, E. (2017). Statistical metamodeling of dynamic network loading. *Transportation Research Procedia*, *23*, 263–282.

Sumalee, A., Zhong, R. X., Pan, T. L., & Szeto, W. Y. (2011). Stochastic cell transmission model (SCTM): A stochastic dynamic traffic model for traffic state surveillance and assignment. *Transportation Research Part B: Methodological*, *45*(3), 507–533.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 27, pp. 3104–3112). Montreal, Canada: Curran Associates, Inc.

Szeto, W. Y., Ghosh, B., Basu, B., & O'Mahony, M. (2009). Multivariate traffic forecasting technique using cell transmission model and SARIMA model. *Journal of Transportation Engineering*, *135*(9), 658–667.

Van Lint, J. W. C., & Hoogendoorn, S. P. (2010). A robust and efficient method for fusing heterogeneous data from traffic sensors on freeways: A method for fusing heterogeneous data from traffic sensors. *Computer-Aided Civil and Infrastructure Engineering*, *25*(8), 596–612.

Wang, D., Cao, W., Xu, M., & Li, J. (2016). ETCPS: An effective and scalable traffic condition prediction system. In J. Pei, Y. Manolopoulos, S. Sadiq, & J. Li (Eds.), *Database systems for advanced applications* (Vol. 9643, pp. 419–436). Cham, Switzerland: Springer.

Wang, J., Gu, Q., Wu, J., Liu, G., & Xiong, Z. (2016). Traffic speed prediction and congestion source exploration: A deep learning method. IEEE 16th International Conference on Data Mining (ICDM), IEEE, Barcelona, Spain, 499–508.

Williams, B., Durvasula, P., & Brown, D. (1998). Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record: Journal of the Transportation Research Board*, *1644*, 132–141.

Williams, J. K., Neilley, P. P., Koval, J. P., & McDonald, J. (2016). Adaptable regression method for ensemble consensus forecasting. AAAI Conference on Artificial Intelligence, Phoenix, Arizona, 3915–3921.

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Xue, Y., & Li, Y. (2018). A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects. *Computer-Aided Civil and Infrastructure Engineering*, *33*(8), 638–654.

Yao, B., Chen, C., Cao, Q., Jin, L., Zhang, M., Zhu, H., & Yu, B. (2017). Short-term traffic speed prediction for an urban corridor. *Computer-Aided Civil and Infrastructure Engineering*, *32*(2), 154–169.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., & Li, Z. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. AAAI Conference on Artificial Intelligence, New Orleans, LA.

Yuan, J., Zheng, Y., Xie, X., & Sun, G. (2013). T-drive: Enhancing driving directions with taxi drivers' intelligence. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 220–232.

Yuan, Y., Van Lint, H., Van Wageningen-Kessels, F., & Hoogendoorn, S. (2014). Network-wide traffic state estimation using loop detector and floating car data. *Journal of Intelligent Transportation Systems*, *18*(1), 41–50.

Zhang, H. M. (2000). Recursive prediction of traffic conditions with neural network models. *Journal of Transportation Engineering*, *126*(6), 472–481.

Zhang, J., He, S., Wang, W., & Zhan, F. (2015). Accuracy analysis of freeway traffic speed estimation based on the integration of cellular probe system and loop detectors. *Journal of Intelligent Transportation Systems*, *19*(4), 411–426.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. AAAI Conference on Artificial Intelligence, San Francisco, CA, 7.

Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016). DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 1–4). New York, NY: ACM Press.

Zhang, X.-L., & Wang, D. (2016). A deep ensemble learning method for monaural speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(5), 967–977.

Zheng, Z., & Su, D. (2016). Traffic state estimation through compressed sensing and Markov random field. *Transportation Research Part B: Methodological*, *91*, 525–554.

How to cite this article: Liu Y, Liu Z, Vu HL, Lyu C. A spatio-temporal ensemble method for large-scale traffic state prediction. *Comput Aided Civ Inf*. 2019;1–19. https://doi.org/10.1111/mice.12459