Neural Process Lett
https://doi.org/10.1007/s11063-018-9804-x

A Learning-Based Multimodel Integrated Framework for Dynamic Traffic Flow Forecasting

Teng Zhou[1] · Guoqiang Han[2] · Xuemiao Xu[2] · Chu Han[3] · Yuchang Huang[4] · Jing Qin[5]

Abstract Accurate and timely traffic flow forecasting is essential for many intelligent transportation systems. However, it is quite challenging to develop an efficient and robust forecasting model due to the inherent randomness and large variations of traffic flow. Over the past two decades, a variety of traffic flow forecasting models have been proposed. While each model has its merits and can achieve satisfactory forecasting results under certain traffic conditions, it is difficult for a single model to deal with various conditions well. In this paper, we proposed a novel deep learning-based multimodel integration framework in order to overcome the limitations of previous methods in dealing with large variations and uncertainties of traffic flow and hence improve the forecasting accuracy. Our framework can dynamically choose an optimal model or an optimal subset of models from a set of candidate models to forecast the future traffic flow conditions according to current input data. We employ stacked autoencoder (SAE), a simple yet efficient deep learning architecture, to extract the implicit relationships hidden in the traffic flow data and employed labeled data to fine tune the parameters of the architecture. Compared with the hand-crafted features and explicable dependence relations leveraged in previous models, the features learning from SAE are more representative and hence have more powerful forecasting capability. In addition, we propose a model-driven scheme to automatically label the training data and develop three strategies to integrate multiple models. Extensive experiments performed on three typical traffic

✉ Xuemiao Xu
xuemx@scut.edu.cn

1 Department of Computer Science, College of Engineering, Shantou University, Shantou, China

2 School of Computer Science and Engineering, South China University of Technology, Guangzhou, China

3 Department of Computer Science and Engineering, The Chinese University of Hong Kong, Sha Tin, Hong Kong

4 College of Mathematics and Information, South China Agricultural University, Guangzhou, China

5 Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Published online: 26 March 2018

flow datasets demonstrate the proposed framework outperforms state-of-the-art models and achieves much more accurate forecasting results under large and sudden variations.

Keywords Traffic flow forecasting · Stacked autoencoder · Multimodel integration · Variation and uncertainty · Deep learning

1 Introduction

Accurate and timely traffic flow forecasting is a crucial prerequisite for many intelligent transportation applications, such as traffic management and control, transportation networks design, and individual transport planning. Accurate traffic flow forecasting not only enables appropriate allocation of transportation resources, but also allows individuals to make better travel planning to save time and avoid traffic congestion. To the end, a lot of researchers have been dedicated to developing various models for efficient traffic flow forecasting.

However, although these existing model has its merits and is applicable to a specific traffic condition, it is difficult for a single model to deal with all conditions due to the significant space inhomogeneity and time-varying characteristics of traffic flow, not to mention traffic accidents occasionally happen on the roads. For example, the historical average models were developed based the periodic characteristic of traffic flow, and thus it was not able to make a proper response to unexpected incidents. On the other hand, the Kalman filter based methods usually suffer from two major limitations: (1) the traffic variations are assumed to be linear; and (2) the traffic states are assumed to be Gaussian distributed, but the traffic states are sophisticated in many real-life situations and the estimating states are not always Gaussian. Some non-linear models were therefore proposed to deal with the non-linearity of traffic flow. The support vector machine regression models (SVR) map the non-linear relations to a high-dimensional space by minimizing the function gap and the empiric risk, while the models based on artificial neural networks leverage activation function to non-linearly map the inputs by minimizing the error between the measurements and the outputs. However, these models either heavily depend on the amount and quality of training data or are so computationally intensive that cannot be applied in practice.

In order to overcome these shortcomings, researchers propose various approaches to enhance these models by preprocessing the traffic flow data, combining the seasonality nature and improving training mechanisms for some learning based methods. However, these improvements based on a single model still could not achieve satisfactory results due to the large variations of traffic flow data in both spatial and temporal domains. For example, in the same week, the traffic flow data in weekday and weekend have quite different characteristics while in the same day the traffic flow data in the morning and in the evening have different features. In addition, there are a lot of factors that will influence the changes of traffic flow, such as weather condition or accidents. In this regard, it is difficult to accurately model the traffic flow data using only one single model.

In this paper, we propose to integrate a set of representative models into a unified ensemble framework and exploit stacked autoencoder networks (SAE) to select an optimal model or an optimal subset of models to perform traffic flow forecasting in a real-time manner according to the current situation. In order to train the SAE, we acquire the training dataset from historical real data and develop a model-driven mechanism to automatically label them. We then train the SAE based on the training dataset and fine-tune the trained model by taking use the probability obtained in the labeling stage. Once the SAE is established, we

can input the current traffic flow and the SAE will choose an optimal model or an optimal subset of models based on the probability distribution. To efficiently and flexibly leverage the probability distribution for forecasting, we implement three strategies to integrate the candidate models according to their probabilities, namely conditional expectation, maximum probability and selective integration. In our implementation, we choose six representative models as the candidates while our framework is extensible to include more models. To validate the effectiveness of the proposed framework, we perform extensive experiments on three representative traffic flow datasets: Netherlands Amsterdam motorways dataset, the dataset of Caltrans Performance Measurement System (PeMS) and the dataset from the Traffic Data Acquisition and Distribution (TDAD) system. Experimental results demonstrate the proposed method outperforms state-of-the-art models and achieves much more accurate forecasting results under large variations and uncertainties by dynamically selecting the optimal model(s).

Our contributions can be summarized as follows:

- to the best of our knowledge, our work is the first attempt to employ deep learning architecture to integrate multiple models for accurate and real-time traffic flow forecasting; thanks to the powerful learning capability of SAE, our method can overcome the limitations of previous methods in dealing with large variations and uncertainties of traffic flow and achieve much better performance compare with existing models;
- we employ SAE, a simple yet efficient deep learning architecture, to extract the implicit relationships hidden in the traffic flow data and employed labeled data to fine tune the parameters of the architecture; compared with the hand-crafted features and explicable dependence relations leveraged in previous models, the features learning from SAE are more representative and hence more powerful to make forecasting; in addition, we propose a model-driven scheme to automatically label the training data to avoid the laborious labeling work, which is an essential prerequisite for many deep learning techniques.

The remainder of this paper is organized as follows. We briefly introduce related works in Sect. 2. Section 3 presents the proposed learning-based multimodel integration framework in details. Experiments and results are reported in Sect. 4. Finally, we draw conclusions in Sect. 5.

2 Literature Review

A variety of traffic flow forecasting models have been proposed in the literature. In this section, we briefly review some models closely related to the proposed framework. Readers can refer to [35] for a more comprehensive review.

Traffic flow forecasting models can be roughly classified into two categories: parametric models and nonparametric models [35]. Parametric models mainly include various time series models, such as linear and non-linear regression models, historical average algorithms [42,44], smoothing techniques [10,33,39,42,57], Kalman filtering methods [5,16,36,40,46,51,58,59] and autoregressive models [2,3,11,17,25,27,34,38,56,57]. We introduce some typical or recently proposed models here. Xie et al. [58] first investigated the application of Kalman filter with discrete wavelet analysis in short-term traffic volume forecasting, which improves the accuracy under large variations of traffic flow by denoising the data with discrete wavelet decomposition and then using the Kalman filter to estimate the weight of the past traffic flow. Ghosh et al. [15] introduced a parsimonious and computationally simple multivariate short-term traffic condition forecasting algorithm using a different

class of time-series models called structural time-series model (STM). Tchrakian et al. [47] described an algorithm for the short-term prediction of traffic with real-time updating based on spectral analysis. Pan et al. [37] extended the SCTM framework to consider the spatial and temporal correlation of traffic flow and to support short-term traffic state prediction. The SCTM is short for stochastic cell transmission model, which describe the macroscopic dynamics of the traffic flow under demand and supply uncertainties [45]. Wang et al. [54] developed an adaptive prediction algorithm for the inflows into the network in regular traffic situations based on an adaptive prediction algorithm of Bohlin [8]. However, although these models are easy to implement and computationally effective, it is difficult for them to resolve the intrinsic model uncertainties.

The nonparametric techniques mainly include nonparametric regression models [13], neural networks [41,53] and support vector regression algorithms [12,19,20,28]. Some typical and newly released methods are briefly introduced here. Boto-Giralda et al. [9] applied wavelet-based denoising self-organizing neural networks to traffic flow forecasting, which learns the self-organizing neural networks with wavelet denoised data. Chan et al. [10] propose a novel neural network (NN) method that employs the hybrid exponential smoothing method and the Levenberg–Marquardt (LM) algorithm for framework NNs. Jeong et al. [23] present a novel prediction model, called online learning weighted support-vector regression (OLWSVR), for short-term traffic flow predictions. Lippi et al. [28] present two new support vector regression models based on the typical traffic flow seasonality. Hu et al. [21] present a hybrid PSO-SVR model, which uses particle swarm optimization (PSO) to search optimal SVR parameters. However, most of these models highly depend on the amount and quality of training data, and hence are not practical in many situations.

In 2006, Hinton and Salakhutdinov [18] described an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis (PCA) as a tool to reduce the dimensionality of data. The key to success of deep learning is mainly due to the ability of high feature extraction using a general-purpose learning procedure [26]. Although there is rare explicit mathematical proof of the theoretical aspects of this success, the common view is that layer-wise training with good criterion may help to learn discriminative features, and a lot of effort has been made [6,48,49]. Recently, Huang et al. [22] and Lv et al. [30] have applied stacked autoencoder and deep belief networks to traffic flow forecasting, respectively. They are the pioneers that introduced deep learning approaches into this field. The normalization operation is required to assign the maximum traffic flow to the forecasting point, i.e. the capacity of the transportation facility can accommodate at that point. However, the capacity varies under different conditions, such as environmental condition, season, degradation of the road surface, and so on. This issue was discussed in previous studies [1,7,32]. In this regard, the undetermined capacity will affect the forecasting accuracy of their deep networks. We suggest readers refer to [28] for more details on approaches for short-term traffic flow forecasting under the common view of probabilistic graphical models.

3 The Proposed Multimodel Integrated Framework

As analyzed, most existing forecasting models cannot well deal with the large variation and inherent uncertainties of traffic flow. To overcome this limitation, we propose a novel learning-based multimodel integration framework, which can automatically and adaptively choose (or construct) a suitable forecasting model from a set of candidate models for arbitrary

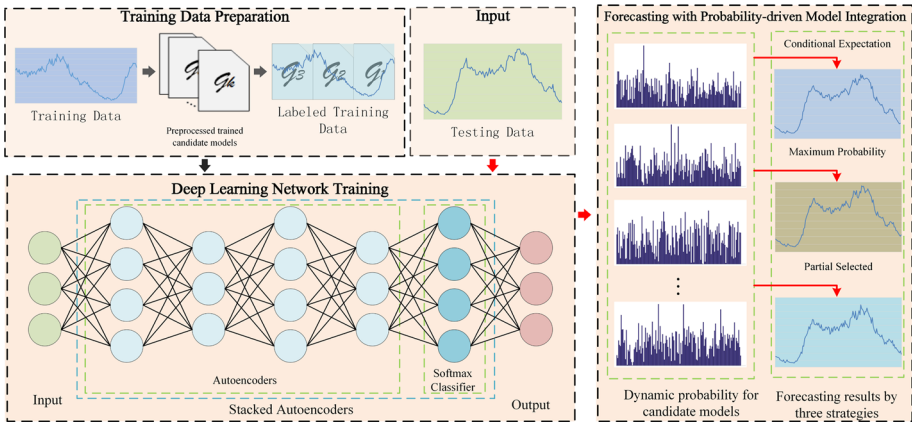


Fig. 1 The flowchart of the proposed framework. The training data are input to all the pre-trained candidate models to perform labeling based on a model-driven mechanism. Then the labeled data are used to train the stacked autoencoder deep network. At the forecasting stage, the testing data are fed into the trained network to determine which model(s) is (are) more suitable to conduct the forecasting under current condition. The results can be calculated by three strategies

input traffic flow. Figure 1 shows the flowchart of the proposed framework, which consists of two phases: the training phase and the forecasting phase.

In the training phase, two key issues are involved: the training data preparation (the left-top in Fig. 1) and the deep learning network training (the left-below in Fig. 1). To prepare the labeled training data for the deep learning network training, an automatic labeling scheme is developed to determine the most suitable forecasting model for each group of training traffic flow by selecting the model with the most accurate forecasting result. When the labeled data are ready, the stacked autoencoder (SAE) algorithm is employed to construct the deep learning network. In the forecasting phase, given the testing traffic flow data as input, the suitability probability for each candidate model can be obtained based on the trained deep network. Then, three strategies, including conditional expectation, maximum probability, and selective integration, are designed to calculate the final forecasting results.

Without loss of generality, we describe our method based on an assumption that we always focus on a certain measurement location in the input road network. Other locations in the road network can be dealt with in the same way.

3.1 Training Data Preparation

To prepare the training data for a certain measurement location, two sets of training data are required. One set is used to train the candidate models in order to determine the optimal parameters for each model. We denote it as $X_{\mathcal{M}} = \{x_m\}_{m=1,\ldots,M}$, where M is the number of groups of traffic flow data in $X_{\mathcal{M}}$. The other set is used to train the deep network for calculating the suitability probabilities of the candidate models for a specific group of traffic flow data. We denote it as $X_{\mathcal{F}} = \{x_n\}_{n=1,\ldots,N}$, where N is the number of groups of traffic flow data in $X_{\mathcal{F}}$. There is no overlap between $X_{\mathcal{M}}$ and $X_{\mathcal{F}}$. Each element (either x_m or x_n) in the dataset corresponds to a group of traffic flow collected at a certain time interval. In practice, the traffic flow data in the coming time intervals are usually associated with the data in the previous continuous time intervals and surrounding locations. In this case, we further define each x as $\{v_{i,j}\}_{i=1,\ldots,O,\, j=t,\ldots,t-R+1}$, where $v_{i,j}$ is the traffic flow measured at location i and

time interval j; R is the number of associated time intervals; O is the number of measurement locations in a road network. Note that usually all of the measurement locations in the road network are involved in the forecasting, since the road network is often so complicated that it is difficult to accurately determine the associated locations for each case.

3.1.1 Candidate Model Training

Our framework is designed to intelligently integrate multiple forecasting models to deal with large variations and random uncertainties of traffic flow. We choose six representative models as the candidate models, which can cover most of the traffic conditions in practice. The six candidate models are the most popular and widely used models for traffic flow forecasting nowadays. Among them, we select four time series models (the historical average (HA) [42], the random walk (RW) [28], the autoregression (AR) [38], and the Kalman filtering (KF) [58]) and two statistics-based learning models (ANNs [61] and SVR [21]). Note that the proposed framework is extensible to include more models.

Before using the candidate models for forecasting, a set of parameters for each candidate model should be determined based on the training data. For example, we need to learn the weights for each layer of ANNs by back-propagation algorithms [61]; for SVR, we need to train the support vector machine by optimizing function margin and expected risk [21]; for Kalman Filtering models, we need to obtain the variances of the measurement noise and the process noise by EM algorithm [58]; for historical average models, we need to compute the historical average traffic flow; for AR, we need to determine the order of the autoregression model.

For a certain measurement location, the training data $X_{\mathcal{M}}$ is prepared for training the candidate models. Then the trained candidate models with optimal parameters are obtained, which can be used for the following forecasting. We indicate the trained kth candidate model as $\mathcal{G}^{(k)}$.

3.1.2 Training Data Labeling

The goal of our framework is to choose the most suitable candidate model for a given group of traffic flow. This requires that the framework training data should be labeled with the most suitable candidate model. Actually, each traffic forecasting model has its own merits and disadvantages [54]. For examples, the historical average approach has a key drawback in responding to unexpected incidents; the Kalman filtering approach is prone to produce overshoots; and the performance of ANNs heavily relies on the amount and quality of training data. However, it is still difficult to determine the most suitable candidate model for a set of training data based on these characteristics, considering the complexity of traffic conditions. In this regard, we propose a simple yet effective scheme to label the training data.

It is reasonable to assume that, for a group of traffic flow, the candidate model which produces the minimum prediction error is most likely to be its most suitable model. According to this assumption, for a certain measurement location o, given the nth group of traffic flow x_n in the dataset $X_{\mathcal{F}}$, the prediction error by the kth candidate model can be computed using the following equation:

$$\epsilon_k = \left| v_{t+1,o} - \mathcal{G}^{(k)}(x_n) \right|, \tag{1}$$

where $v_{t+1,o}$ is the traffic flow data collected at location o at time $t+1$, $\mathcal{G}^{(k)}(x_n)$ is the predicted value of $v_{t+1,o}$ by the candidate model k, and ϵ_k is the prediction error of model k.

Thus, the model which produces the minimum prediction error can be obtained by

$$k^* = argmin_{k=1,\ldots,K}(\epsilon_k), \tag{2}$$

where K is the number of candidate models, and k^* is the index of the most suitable candidate model for the nth group of traffic flow in $X_{\mathcal{F}}$. That means the nth group of traffic flow x_n can be labeled with $y_n = k^*$. Thus, for a certain measurement location in a road network, its labeled training data can be obtained and indicated as $Y_{\mathcal{F}} = \{x_n, y_n\}_{n=1,\ldots,N}$.

3.2 Deep Learning Network Training

With labeled training data, we target for training a classifier that can choose the most suitable candidate model for each group of traffic flow. Classification algorithms based on machine learning have made great progress in images processing as well as other pattern recognition tasks [4]. Many of them need to design hand-crafted features, which are capable of differentiating the target categories. However, in our application, it is difficult to use common features, such as occupancy or average speed, or design other hand-crafted features to differentiate the forecasting models effectively due to the large variations of traffic flow states.

Recently, deep learning [26] has drawn a lot of academic and industrial interests [14,31, 55], which can automatically discover the implicit relationships inside the data in a hierarchical manner. This feature motivates us to exploit it to reveal the complicated relationships of traffic flow data collected at different measurement points and different time slots in a road network in order to achieve more accurate forecasting based on these implicit relationships. In other words, it has great potential to learn representative high-level spatiotemporal features, that cannot be expressed with traditional hand-crafted descriptors, in order to find suitable models for more accurate forecasting. Moreover, deep learning can achieve the real-time forecasting, since the time-consuming training process can be conducted off-line.

Currently, lots of models [26] have been proposed for deep learning. We employ the stacked autoencoder (SAE), a simple yet efficient deep learning network architecture, in our application. As shown in the left-below part of Fig. 1, the SAE is a neural network consisting of multiple layers of autoencoders, which learn the features in an unsupervised way with the unlabeled data. A classifier is then integrated to further fine-tuning the whole network to achieve accurate classification with labeled data.

3.2.1 Autoencoder

One of the main features of autoencoders is that they can be used for big data analysis, as both the unsupervised training and the fine-tuning scale linearly in time and space with respect to the number of training cases [18]. To the end, it is quite suitable for traffic flow forecasting, which is usually involved a large amount of data. As shown in Fig. 2, each autoencoder is a neural network with only one input layer, one hidden (feature) layer and one output layer, and the the output is encouraged to reproduce the input [50]. Suppose that the input is an unlabeled dataset X, the hidden layer H and the output layer Z can be formulated as:

$$\begin{aligned} H &= f(W^{(1)}X + b^{(1)}), \\ Z &= f(W^{(2)}H + b^{(2)}). \end{aligned} \tag{3}$$

where $f(\cdot)$ is a non-linear mapping function. Recent studies found that $f(x) = \frac{1}{1+\exp(-x)}$, $f(x) = \tanh(x)$ or $f(x) = max(0, x)$ achieve better results. In this paper, we set $f(x) = \frac{1}{1+\exp(-x)}$. Note that in our case, X is the traffic flow data $X_{\mathcal{F}}$ and H is the learned feature

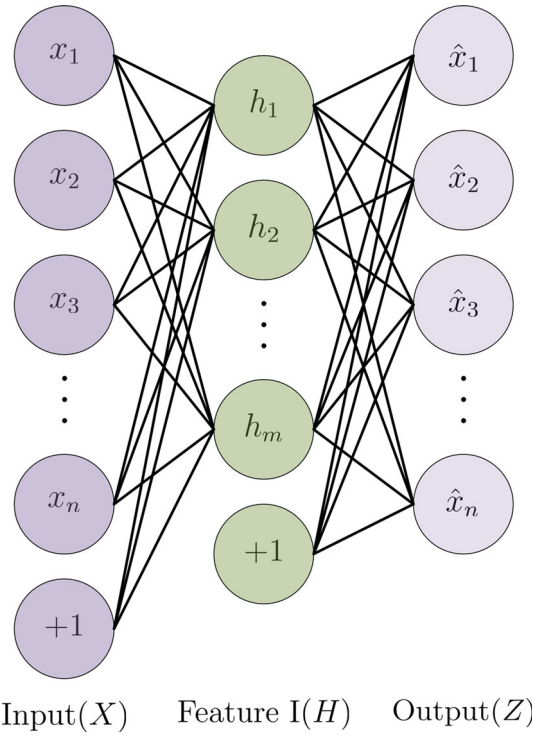


Fig. 2 Illustration of an autoencoder. The input data $X = X_{\mathcal{F}}$ is a matrix in which each column corresponds to a group of traffic flow. Correspondingly, H is a matrix in which each column corresponds to the feature vector for a group of traffic flow as input, and Z is a matrix in which each column corresponds to a group of reconstructed traffic flow. Here, we only show one column corresponding to one group of traffic flow for simplification

of the traffic flow data. The $\{W^{(1)}, b^{(1)}\}$ and $\{W^{(2)}, b^{(2)}\}$ are the coefficients for determining the networks from X to H and from H to Z respectively. They can be solved by minimizing the following energy function [50]:

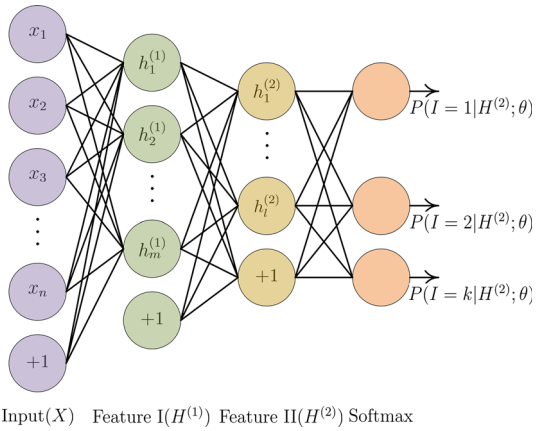
$$E = \frac{1}{2}(X - Z(X))^2 + \frac{\lambda}{2}\left(\left\|W^{(1)}\right\|_F^2 + \left\|W^{(2)}\right\|_F^2\right) + \beta \sum_{j=1}^{s} \mathrm{KL}(\rho||\hat{\rho}_j). \tag{4}$$

where $\|A\|_F$ is the Frobenius norm of matrix A; $KL(p||q)$ is the Kullback–Leibler divergence.

$\hat{\rho}_j$ is the average activation of the jth row of matrix H, defined as $\hat{\rho}_j = \frac{1}{N}\sum_{i=1}^{N} H_{i,j}$ where N is the number of groups of traffic flow in the input $X_{\mathcal{F}}$; ρ is the target sparsity; s is the dimension of H; λ and β are control parameters.

This energy function is designed to achieve three objectives which exactly correspond to three components on the right side of Eq. 4. The first one is to minimize the difference between the input data X and the reconstructed data Z. In order to prevent over-fitting, a regularization term is introduced for decreasing the magnitude of the weights. Lastly, sparsity constraints of Kullback-Leibler divergence is employed to extract more representative features, which can better differentiate the categories of the input data, since there are lots of correlations among the traffic flow data.

Fig. 3 Illustration of SAE



3.2.2 Stacked Autoencoder

The SAE is a neural network consisting of multiple layers of autoencoders for feature extraction with the unlabeled data, and a classifier for fine-tuning the whole network with labeled data. Hence, the training of the SAE is composed of two steps: (1) unsupervised learning for feature extraction using layer-wise autoencoders; and (2) supervised fine-tuning for accurate classification with a softmax classifier.

Feature Extraction In the SAE, the hidden layer of the previous autoencoder would be the input of the following autoencoder. For example, as shown in Fig. 3, the hidden layer $H^{(1)}$, which is the feature extracted by the 1st autoencoder, is the input of the 2nd autoencoder. We rewrite Eq. 3 as the following equation [52]:

$$\begin{aligned} H^{(l)} &= f(W^{(l,1)}H^{(l-1)} + b^{(l,1)}) \\ Z^{(l)} &= f(W^{(l,2)}H^{(l)} + b^{(l,2)}) \end{aligned} \tag{5}$$

where $H^{(l-1)}$, $H^{(l)}$ and $Z^{(l)}$ is the input, hidden and output layer of the lth autoencoder, respectively. Then we process the layer-wise training greedily based on Eq. 4 with $H^{(0)} = X_{\mathcal{F}}$. In this case, the coefficient matrices for each autoencoder $\{W^{(l,1)}, b^{(l,1)}, W^{(l,2)}, b^{(l,2)}\}$ can be trained. In our application, based on the trained network of the multi-layer autoencoders, we can obtain the different layers of features $H^{(l)}$ for the input traffic flow $X_{\mathcal{F}}$. Although these features can implicitly reflect the relationships of the input traffic flow data, we should fine tune the network parameters with labeled data to enhance the representation capability of these features so that accurate forecasting can be achieved.

Network Fine-Tuning With the feature automatically extracted from the traffic flow, we need a classifier to fine-tune the whole network to evaluate the suitability of a candidate model for a specific input traffic flow. To the end, we attach a classifier at the end of multi-layer autoencoders for fine-tuning the whole network through backpropagation.

In our implementation, we apply a generalization version of a logistics regression classifier, softmax classifier [24], for multi-classification. Given the feature $H^{(L)}$ extracted from the last autoencoder in the network, where L is the number of layers of autoencoders, the condition probability of $H^{(L)}$ can be denoted as [52]:

$$\begin{bmatrix} P(I=1|H^{(L)};\theta) \\ P(I=2|H^{(L)};\theta) \\ \vdots \\ P(I=K|H^{(L)};\theta) \end{bmatrix} = \frac{1}{\sum_{k=1}^{K}\exp(\theta^{(k)\top}H^{(L)})} \begin{bmatrix} \exp(\theta^{(1)\top}H^{(L)}) \\ \exp(\theta^{(2)\top}H^{(L)}) \\ \vdots \\ \exp(\theta^{(K)\top}H^{(L)}) \end{bmatrix} \tag{6}$$

where I denotes the index of a candidate model; θ is the coefficient matrix for the classifier indicating as $\theta = \left[\theta^{(1)}\ \theta^{(2)}\ \cdots\ \theta^{(K)}\right]$; $\theta^{(k)}$ is the coefficient vector indicating the contribution of each element in $H^{(L)}$ to the kth candidate model. Thus, $P(I=k|H^{(L)};\theta)$ denotes the probability of $H^{(L)}$ if it is classified to the kth candidate model based on coefficient matrix θ. The term $\frac{1}{\sum_{k=1}^{K}\exp(\theta^{(k)\top}H^{(L)})}$ is used for normalization.

The cost function of softmax can be derivated as:

$$\begin{aligned} J(\theta) &= -\left[\sum_{n=1}^{N}\sum_{k=1}^{K}\delta_{y^{(n)},k}\log\frac{\exp(\theta^{(k)\top}H_n^{(l)})}{\sum_{k=1}^{K}\exp(\theta^{(k)\top}H_n^{(l)})}\right] \\ \delta_{y^{(n)},k} &= \begin{cases} 0, & if \quad y^{(n)} \neq k \\ 1, & if \quad y^{(n)} = k \end{cases} \end{aligned} \tag{7}$$

where $H_n^{(l)}$ is the feature extracted by the lth autoencoder when the nth group of traffic data $x_{\mathcal{F}}^{(n)}$ is given as input. In this case, $H_n^{(l)}$ is actually determined by $\{W^{(l,1)}, b^{(l,1)}\}$ for each layer of autoencoder. We employ the Kronecker delta δ to only focus on the candidate model which is exactly the same as the labeled one in $Y_{\mathcal{F}}$. Through maximizing the probability of all groups of traffic flow in $X_{\mathcal{F}}$ when they are classified into the labeled candidate models, the best classification can be obtained. We utilized the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm [29], a typical gradient descent algorithm to solve this cost function.

3.3 Forecasting by Probability-Driven Model Integration

For a certain measurement location, once its deep network is trained with the labeled training data $Y_{\mathcal{F}}$, given a group of testing data $x_{\mathcal{T}}$, the probability for each candidate model can be obtained, as shown in the right part of Fig. 1. The probability $P(I=k|x_{\mathcal{T}})$ indicates the suitability of the kth candidate model for the input $x_{\mathcal{T}}$. In order to flexibly and comprehensively leverage the probabilities for better forecasting, we introduce three strategies to integrate the candidate models according to their probabilities.

3.3.1 Conditional Expectation

The most straightforward way is to calculate the conditional expectation of the forecasting result according to the probabilities. The conditional expectation can be calculated as:

$$\hat{v}_{o,t_{\mathcal{T}}+1} = \begin{bmatrix} \mathcal{G}^{(1)}(x_{\mathcal{T}}) \\ \mathcal{G}^{(2)}(x_{\mathcal{T}}) \\ \vdots \\ \mathcal{G}^{(K)}(x_{\mathcal{T}}) \end{bmatrix}^{\top} \begin{bmatrix} P(I=1|x_{\mathcal{T}}) \\ P(I=2|x_{\mathcal{T}}) \\ \vdots \\ P(I=K|x_{\mathcal{T}})) \end{bmatrix} \tag{8}$$

where $\hat{v}_{o,t_{\mathcal{T}}+1}$ is the forecasted traffic flow at location o in next time point; $\mathcal{G}^{(k)}(x_{\mathcal{T}})$ is the forecasting result of candidate model k; and $P(I=k|x_{\mathcal{T}})$ is the suitability probability of candidate model k for data $X_{\mathcal{T}}$. This strategy works well in most of the cases in our

experiments, but it may fail when one or more candidate models produce forecasting results with large deviations, which lead to relatively big errors in the calculation of conditional expectation.

3.3.2 Maximum Probability

To avoid the effects of unsuitable candidate models on the forecasting result, the second strategy is proposed to only consider the candidate model with maximum probability, which is defined as:

$$k^* = arg\max_k P(I = k|x_{\mathcal{T}}). \tag{9}$$

where k^* is the index the model with maximum probability. Thus, the forecasting result can be calculated as:

$$\hat{v}_{o,t_{\mathcal{T}}+1} = \mathcal{G}^{(k^*)}(x_{\mathcal{T}}). \tag{10}$$

This strategy is more efficient and can achieve good forecasting results when the probability of the selected model is dominant over all other models. However, when one or more other models' probabilities are close to that of the selected model, the strategy may produce inaccurate results.

3.3.3 Selective Integration

In the third strategy, we propose to consider models with relatively high probabilities instead of only employing the one with maximum probability. To determine whether a candidate model should be considered or not, we define a threshold value ψ^*, and calculate the value of $\psi = \frac{P(k|x_{\mathcal{T}})}{P(k^*|x_{\mathcal{T}})}$ for each model. When $\psi \geq \psi^*$, the kth model is chosen for forecasting. Otherwise, it is ignored. After the selected models are determined, we figure out the forecasting result by re-normalizing their probabilities and calculating the conditional expectation of these selected models according to Eq. 8. In essence, it is a trade-off scheme of conditional expectation strategy and maximum probability strategy.

Note that our framework can achieve real-time forecasting for arbitrary input traffic flow, as the forecasting phase performed online only involves simple matrices addition and multiplication. The computation-intensive training phase is performed off-line.

4 Cases Study

We validate the proposed learning-based multimodel integrated framework on three real traffic flow datasets. We compare our method with commonly used forecasting models, as well as the three strategies equipped in our framework, in terms of two frequently used criteria: the root mean square error (RMSE) and the mean absolute percentage error (MAPE).

4.1 Data Description

The first dataset was collected from four motorways by Wang et al. [54], namely A1, A2, A4, and A8, which end on the ring road of Amsterdam (the A10 motorway), as shown in Fig. 4. We simply depict the four motorways as follows.

– The A1 motorway connects the city of Amsterdam with the German border, which is also a European route. The European route E30 follows the A1 motorway from the

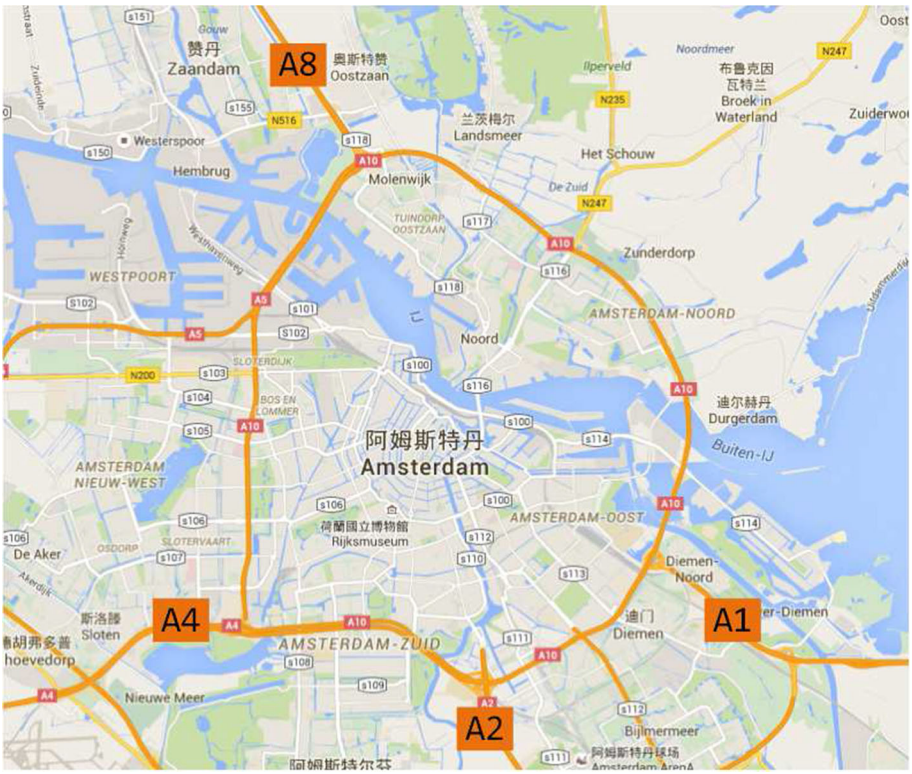


Fig. 4 The four motorways namely A1, A2, A4, and A8, which end on the ring road of Amsterdam. The four measurement sites are without highly correlations spatially

interchange Hoevelaken in the Netherlands. There is the first high-occupancy vehicle (HOV) 3+ barrier-separated lane in Europe on A1 motorway. The traffic flow in this HOV lane dramatically changes over time, making the forecasting quite challenging.

– The A2 motorway is one of the busiest highways in the Netherlands, which connects the city of Amsterdam and the Belgian border. In our experiment, we use the data collected in 2010 before the motorway is widened to examine if the proposed framework can perform well with congestions.
– The A4 motorway is part of the Rijksweg 4, which starts from Amsterdam to the Belgian border. The A4 motorway has priority from the eastern direction until the interchange De Nieuwe Meer, then travels to the southeast.
– The A8 motorway starts from the A10 motorway at interchange Coenplein, ends at Zaandijk less than 10 km.

The four measurement sites locate on the motorways a short distance before the merge points to the ring road. The data were provided from May 20, 2010, to June 24, 2010, collected by MONICA sensor. The raw data were aggregated by vehicles per hour in 1 min. We aggregate the raw data into 5- and 10-min. The 5/10-min aggregation of this dataset is the average vehicles per hour in these 5/10 min, which is in a consensus with the previous studies [54,60], who also used this dataset. We aim at employing the first dataset to demonstrate

the performance of the proposed framework for a dataset with similar patterns but without high correlations.

The second group of traffic flow data was acquired from the performance measurement system (PeMS) deployed by California Department of Transportation. PeMS provides access to real-time and historical data (since 1999) in various formats. These data were collected from over 35,000 detectors with an interval of 30 s.

In our experiments, we employed the traffic flow data of the first 9 weeks in 2015, i.e. Jan 5, 2015, to Mar 8, 2015, from North Center of California. The data are aggregated in 5-min granularity and acquired from 1254 individual measurement stations. In this dataset, the raw data are aggregated by vehicles in 5 min, and the same as the third dataset. Huang et al. [22], Lv et al. [30] and Xie et al. [58] also make the same aggregation on these datasets. Note that as 61 stations were suffering from hardware failure during this period, we utilized data from 1193 stations indeed.

The third dataset was obtained from the traffic data acquisition and distribution (http://www.its.washington.edu/tdad/) database. These data were collected from four different detectors (ES-088D, ES-855D, ES-708D, and ES-645D) located on Interstate 5 (I-5), Interstate 90 (I-90), and Interstate 405 (I-405) in Seattle. The dataset has 35 days of traffic data (May 30, 2005–July 3, 2005). The raw 24-hour traffic flow data were collected every 20 s and then aggregated in 5- or 10-min granularity.

4.2 Candidate Model Configuration

We took six widely-used forecasting models, namely historical average model (HA) [28], random walk model (RW) [28], auto regression (AR) [38], ϵ-support vector machine regression model (ϵ-SVR) [21], Kalman filtering model [38], and artificial neural network (ANN) [61], as the candidate models in our framework. Note that the proposed framework is extensible to integrate more forecasting models, but we think the above-mentioned six models can cover most traffic situations. For each of the three experimental datasets, we employed parts of the data to train the candidate models. The concept of the candidate models is simply introduced as follows.

Historical Average Model This model predicts for a given time of the day the average of the same time on the same day in previous weeks.

Random Walk Model This model simply predicts the traffic flow next moment as equal to the current condition.

Autoregression Model The autoregression model is a representation of a random process and it has been widely used in traffic flow forecasting due to the randomness of the traffic flow. In the autoregression model with order p, the current traffic flow is represented by a weighted combination going back p periods, following a random disturbance in the current period. In this regard, the order p is critical for the autoregression models. If the order is low, the valuable information contained in the lagged will be omitted. On the other hand, if the order is too high, more coefficients need to be estimated, and additional errors will consequently be introduced. The order in our experiment is set to 8 by cross-validate of our training data.

ϵ-Support Vector Machine Regression Model For the ϵ-Support Vector Machine Regression Model, several parameters need to be set beforehand. The regression horizon is set the same as the autoregression model. The two weeks traffic flow measurements are aggregated 5-min and 10-min granularity to make up the training matrices. We use radial basis function (RBF) as the kernel type in this study. The cost parameter C is set to the maximum difference

Table 1 Configurations of ANN model

Parameters	Values
Hidden layers	1
Goal	0.001
Spread	2000
MN	40
DF	Default

between the traffic flows. The width parameter γ for the RBF kernel is set to 3×10^{-6}. The ϵ-insensitive loss for ϵ-SVR is fixed to 1 in this study.

Kalman Filtering Model Since the raw traffic flow data have too many fluctuations for the direct Kalman filter to handle, we introduce a wavelet denoising procedure proposed by Xie et al. [58]. We use Daubechies 4 as the mother wavelet. Different from Xie et al. [58], we simply set the variance of the process error Q as a small value, namely $0.1 \times \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The variance of the measurement noise is considered as 0. The initial state of the dynamic system is set to $\left[\frac{1}{n}, \ldots, \frac{1}{n}\right]$, where n is set to 8, the same as Xie et al. [58]. The initial state estimation error covariance matrix is $10^{-2} \times \mathbf{I}$.

Artificial Neural Network Model We employ the artificial neural networks introduced in Zhu et al. [61]. The network parameters are described in Table 1, where most of them are consistent with [61].

4.3 Experimental Setup

In the first dataset of Amsterdam motorways, the collected data are divided into three parts, the first two weeks are used for candidate training, the third and fourth are used for framework training, and the rest one is used for evaluation. We manually check the raw data to remove the incorrect data, i.e. the value is -1, caused by the hardware failure.

The architecture of SAE used in the first dataset is set to [120, 60, 30]. The scaling parameter for l_2 weight regularization penalty in Eq. 4 is set to 0.1. The sparsity is set to 0.03. We randomly drop out some measurements of the training inputs to improve the network performance (see [43] for more details). The relative threshold ψ in the strategy 3 is set to 0.7.

In the dataset of PeMS, the data of the first four weeks are used to train the candidate models, and the data of the following four weeks are used to train the proposed framework, the data of the last week are used to evaluate the performance of the proposed framework. The architecture of stacked autoencoder network in this experiments is set to [1200, 700, 400]. The scaling parameter for l_2 weight regularization penalty in Eq. 4 is set to 0.05. The sparsity is set to 0.05. We also take the drop out strategy [43] to avoid overfitting.

In the dataset of TDAD, the five weeks data are divided into three parts, the first two weeks are used for candidate models training, the third and fourth are used for framework training, and the rest one is used for evaluation. The deep architecture of stacked autoencoder network in this experiments is set to [100, 40, 20]. Other parameter configurations are the same with the first dataset.

Table 2 The forecasting results of the proposed framework and the candidate models on dataset of Amsterdam motorways

		A1		A2		A4		A8	
		5 min	10 min	5 min	10 min	5 min	10 min	5 min	10 min
SVR	RMSE	303.02	329.09	221.38	259.74	230.04	253.66	162.55	190.30
	MAPE	14.26	14.34	12.31	12.22	13.45	12.23	14.18	12.48
HA	RMSE	399.41	404.84	332.35	348.96	341.97	357.85	213.27	218.72
	MAPE	19.35	16.87	17.69	15.53	18.90	16.72	19.31	16.24
RW	RMSE	325.44	312.92	217.57	223.82	227.66	230.01	163.38	174.14
	MAPE	14.80	12.65	12.66	11.43	13.46	12.06	15.43	12.37
AR	RMSE	292.19	301.44	200.98	214.22	209.80	226.12	151.47	166.71
	MAPE	14.44	13.57	12.20	11.59	13.24	12.70	14.32	12.71
ANN	RMSE	291.25	299.64	200.32	212.95	209.42	225.86	151.30	166.50
	MAPE	13.77	12.61	11.66	10.89	12.98	12.49	14.16	12.53
KF	RMSE	333.27	332.03	217.43	239.87	230.98	250.51	171.36	187.48
	MAPE	14.60	12.46	12.33	10.72	14.50	12.62	15.32	12.63
Strategy1	RMSE	**220.13**	**220.10**	**162.06**	**166.13**	**166.24**	**171.15**	**107.06**	**120.28**
	MAPE	9.55	8.09	8.72	7.23	8.78	7.46	8.70	7.53
Strategy2	RMSE	230.87	234.34	178.50	189.24	176.15	176.68	109.7	123.68
	MAPE	9.32	7.86	8.54	6.91	8.44	7.37	8.24	7.34
Strategy3	RMSE	224.54	224.26	171.27	178.77	172.39	171.43	108.63	120.37
	MAPE	**9.18**	**7.70**	**8.37**	**6.85**	**8.34**	**7.21**	**8.28**	**7.21**

Bold represents the lowest RMSE or MAPE value

4.4 Evaluation Criteria

Two frequently used criteria are employed to evaluate the performance of the proposed approach. The root mean square error (RMSE) measures the average differences between the predictions of a model and measurements of the system being modeled. The mean absolute percentage error is the percentage expression of the differences. The two criteria are defined in Eqs. 11 and 12, respectively:

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\hat{v}(m) - v(m))^2} \tag{11}$$

$$MAPE = \frac{1}{M}\sum_{m=1}^{M}\left|\frac{\hat{v}(m) - v(m)}{v(m)}\right| \times 100\% \tag{12}$$

where $\hat{v}(m)$ and $v(m)$ are the prediction and the true measurement of the mth group of data.

4.5 Results and Discussion

We compare the forecasting results of the proposed learning-based integration framework with those of the candidate models, which are, as mentioned, the most widely used models in traffic flow forecasting. Furthermore, we also provide the results of the three strategies equipped with the framework. The forecasting results of the three datasets are shown in

Tables 2, 3 and 4, respectively. It is observed that, for all the three datasets, our framework achieve more accurate results than the candidate models.

For the dataset of Amsterdam motorways, conditional expectation strategy achieves the lowest RMSE in all measurement locations in terms of both 5-min average aggregation and 10-min average aggregation, while the selective integration strategy achieves the lowest MAPE in terms of both 5-min average aggregation and 10-min average aggregation. The 5- or 10-min average aggregation is the average vehicles per hour in 5 or 10 min, respectively, which is in consensus with the previous studies [54,60]. Note that the results of our framework, regardless of which strategy is employed, are much better than those of the candidate models. For example, compared with the RMSEs of ANN, which achieved the best results among the candidate models, the RMSEs of our conditional expectation strategy decrease 26.5, 21.9, 24.2 and 27.7% at A1, A2, A4, and A8, respectively. The results demonstrate the proposed framework can better deal with the large variations of traffic flow, which is one of the main challenges of traffic flow forecasting, than most existing models. This is because the representation learning scheme (i.e., the SAE) equipped in our framework can implicitly yet effectively estimate the capability of a candidate model in forecasting the traffic flow under the current condition, and dynamically integrate several (or all) models to achieve more accurate results. It is worthwhile to note that, in most cases, the performance of 10-min average aggregated traffic flows are better than that of 5-min average aggregated ones, because the 10-min aggregation has fewer fluctuations. Since the longer time period is extended, the fewer fluctuations the average traffic flow suffers. Although the extension of forecasting time period increases the difficulty of the forecasting, the noises are also reduced by the average operation with the extending period.

For the dataset of PeMS, Table 3 presents the performance of top 10 busiest measurement stations from 1254 individual ones. It is observed from Table 3 that our framework outperforms the candidate models in terms of both RMSE and MAPE. Among the three strategies, in most cases, the selective integration strategy achieves the best results, demonstrating the effectiveness of this strategy in handling dataset with a large number of measurement locations.

The experimental results of TDAD dataset are listed in Table 4. Compared with the candidate models, our framework achieves much better results in all four detection locations in terms of both 5-min aggregation and 10-min aggregation, demonstrating the effectiveness of the proposed framework. Note that the concepts of 5-min aggregation and 10-min aggregation here are different from the concepts in the dataset of Amsterdam motorways. The aggregation strategy of the dataset from Amsterdam is the average vehicles per hour in the last 5 or 10 min, which is the same as [54,60]. The aggregation strategy of the second and third dataset is the total vehicles within 5 or 10 min, which is the same as [9,22,30,58]. In this regard, the RMSEs of 5-min aggregations are smaller than those of 10-min aggregations on the second and the third dataset.

Comparing the forecasting performance of the strategies, in the first dataset, when the traffic flow is low, especially late at night or early in the morning, small prediction errors cause a large relative error, as a result, making strong impacts on the MAPE but without strong impacts on the RMSE. The strategy 1 may integrate some unsuitable models to degrade the MAPE, especially at these moments, so the MAPE of strategy 1 is not as good as strategy 3. However, these small errors have very few impacts on the RMSE. The following reasons may account for the phenomenon that the performances of the three strategies cannot reach a consensus. First, the data of the first dataset are aggregated in vehicles per hour, so the corresponding values are large than that of PeMS and TDAD. Second, the data of the first dataset are without high correlations, while some data in the PeMS dataset are collected from

Table 3 The performance of the proposed framework and the candidate models on PeMS dataset

	Station ID	317166	313393	316387	314433	316593	317910	313386	313671	314821	312562
SVR	RMSE	24.85	34.72	37.76	27.20	17.89	28.46	24.42	25.33	35.90	30.28
	MAPE	4.83	6.26	6.50	6.58	5.91	8.42	6.62	6.75	7.96	8.73
HA	RMSE	26.62	62.96	68.51	52.96	26.88	40.27	30.35	29.74	59.61	41.72
	MAPE	5.13	12.03	13.30	14.35	11.76	11.41	10.57	9.93	12.29	16.49
RW	RMSE	31.55	40.09	43.26	31.55	20.07	33.07	27.46	28.33	41.97	33.64
	MAPE	6.14	6.87	7.27	7.42	6.43	9.78	7.21	7.36	8.91	9.24
AR	RMSE	24.95	36.19	38.65	28.07	18.41	29.13	25.21	26.14	37.75	30.76
	MAPE	4.86	7.01	7.36	6.79	6.16	8.63	6.91	7.01	8.61	8.94
ANN	RMSE	24.95	36.05	38.48	28.01	18.40	29.02	25.14	26.11	37.67	30.73
	MAPE	4.86	6.97	7.25	6.74	6.13	8.61	6.90	6.97	8.53	8.83
KF	RMSE	30.57	43.12	47.10	31.14	20.18	32.62	28.08	28.33	44.05	34.79
	MAPE	5.97	7.53	8.06	7.44	6.60	9.62	7.33	7.36	9.39	9.65
Strategy 1	RMSE	20.95	**26.53**	27.95	**21.04**	13.81	20.28	18.20	18.63	27.32	**21.37**
	MAPE	3.84	4.49	4.60	5.02	4.50	5.45	4.81	4.82	5.35	5.70
Strategy 2	RMSE	20.93	28.01	**26.58**	21.82	13.87	20.61	18.28	18.62	**27.31**	20.78
	MAPE	3.67	4.39	**4.26**	5.01	4.27	5.25	4.58	4.63	5.06	5.43
Strategy 3	RMSE	**20.50**	27.72	27.07	21.34	**13.69**	**20.26**	**17.89**	**18.37**	27.34	20.54
	MAPE	**3.64**	**4.35**	4.29	**4.91**	**4.21**	**5.24**	**4.49**	**4.62**	**5.03**	**5.39**

Bold represents the lowest RMSE or MAPE value

Table 4 The performance of the proposed framework and the candidate models on TDAD dataset

		ES088D		ES855D		ES645D		ES708D	
		5 min	10 min	5 min	10 min	5 min	10 min	5 min	10 min
SVR	RMSE	21.85	38.23	18.85	34.47	15.84	24.76	13.85	22.15
	MAPE	9.42	7.76	13.98	11.72	11.55	9.02	10.99	8.74
HA	RMSE	25.74	47.63	23.99	47.25	19.10	36.11	17.04	29.51
	MAPE	10.19	9.42	17.21	16.15	13.18	12.97	12.39	11.05
RW	RMSE	24.71	43.13	20.30	38.43	18.80	29.78	17.01	25.45
	MAPE	10.79	8.60	15.21	13.20	13.41	10.53	12.78	9.95
AR	RMSE	22.12	39.23	19.19	36.66	16.12	27.72	14.22	23.02
	MAPE	9.68	8.35	14.80	14.17	12.08	10.65	11.44	9.50
ANN	RMSE	22.10	39.21	19.14	36.35	16.09	27.68	14.21	23.02
	MAPE	9.60	8.31	14.27	13.01	11.92	10.35	11.35	9.51
KF	RMSE	23.88	41.88	20.91	41.52	17.34	31.47	15.33	25.89
	MAPE	9.92	8.74	14.83	13.29	12.38	10.38	11.62	9.83
Strategy1	RMSE	15.21	**33.05**	11.56	28.65	10.04	**22.95**	9.50	**19.38**
	MAPE	5.79	5.96	7.84	8.67	6.17	7.01	6.46	6.62
Strategy2	RMSE	14.66	33.70	11.30	28.61	**9.63**	23.35	9.01	19.60
	MAPE	**5.42**	5.90	**7.42**	**8.35**	**5.75**	6.87	**5.94**	6.44
Strategy3	RMSE	**14.64**	33.40	**11.29**	**28.57**	9.64	23.22	**8.99**	19.44
	MAPE	5.45	**5.84**	7.43	8.36	5.76	**6.84**	5.96	**6.38**

Bold represents the lowest RMSE or MAPE value

adjacent stations. Third, the data of the first dataset are collected from the freeways, while the data of the second dataset are collected from the freeways and the ramps, and the data from the third dataset are collected from the interstates. Although the performances vary from dataset to dataset, we find some common ground. The MAPE of strategy 3 is slightly better than that of strategy 1 and strategy 2 on the first and second dataset. On the third dataset, the strategy 3 outperforms 3/4 of the 10-min forecasting by the MAPE, and the rest one is very close to the winner. We can draw a conclusion that selective integration can help to reduce mean absolute percent error.

In order to explore the working mechanism of the proposed framework, we take measurement location the A2 as an example, and collect the prediction errors (Fig. 5a–f) and corresponding suitability probabilities figured out from our framework (Fig. 5g–l) of all candidate models. Among them, Fig. 5a, b are the results of two weekend days while others are the results of workdays. It is observed that the prediction errors of all candidate models vary so greatly from time to time that we cannot find a single model has been keeping outperforming others during a week. Each model has its suitable periods with low prediction errors. For example, when the traffic flow keeps stable at noon, the RW model usually achieves better results than other models, while during the racing time in the morning, the KF model achieves smaller prediction errors than others. In this case, in order to achieve more accurate forecasting results, we can choose suitable models for a specific moment to counteract the great variations of traffic flows based on the suitability probabilities figured out from our learning-based framework. As shown in Fig. 5g–l, the framework can dynamically assign different suitability probabilities to each model at every moment. It is observed that the framework assign high probabilities to KF model and SVR model when the traffic

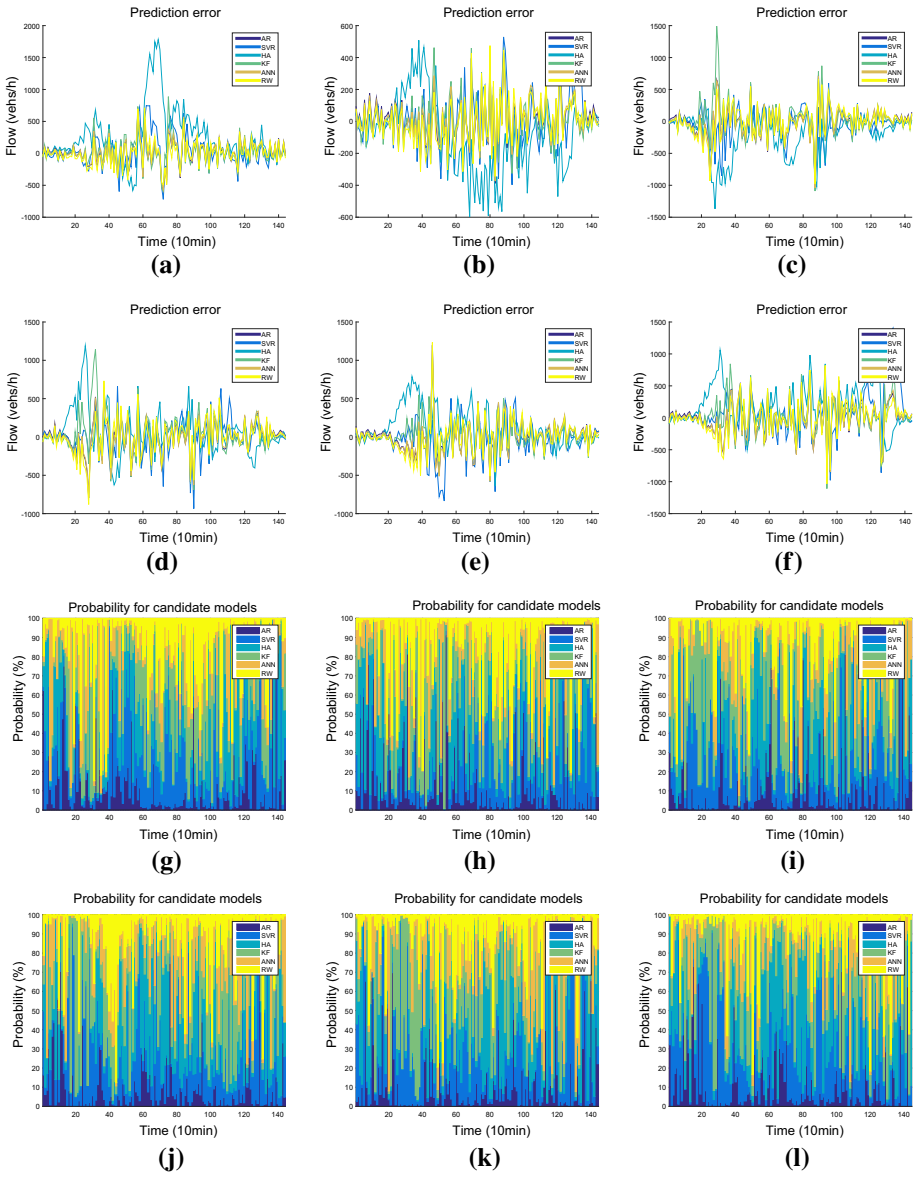


Fig. 5 **a–f** The predication errors of candidate models in a week at measurement location A2 in the dataset of Amsterdam motorways, and **g–l** the corresponding suitability probabilities

flow increases or decreases dramatically while the RW model has high probabilities when the traffic flow is stable. Thus, the proposed learning-based framework can always achieve relatively accurate forecasting results.

Finally, we report several typical yet challenging scenarios to demonstrate the effectiveness of the proposed framework in dealing with variations and uncertainties of traffic flow. In these scenarios, a single model is difficult to achieve satisfactory forecasting.

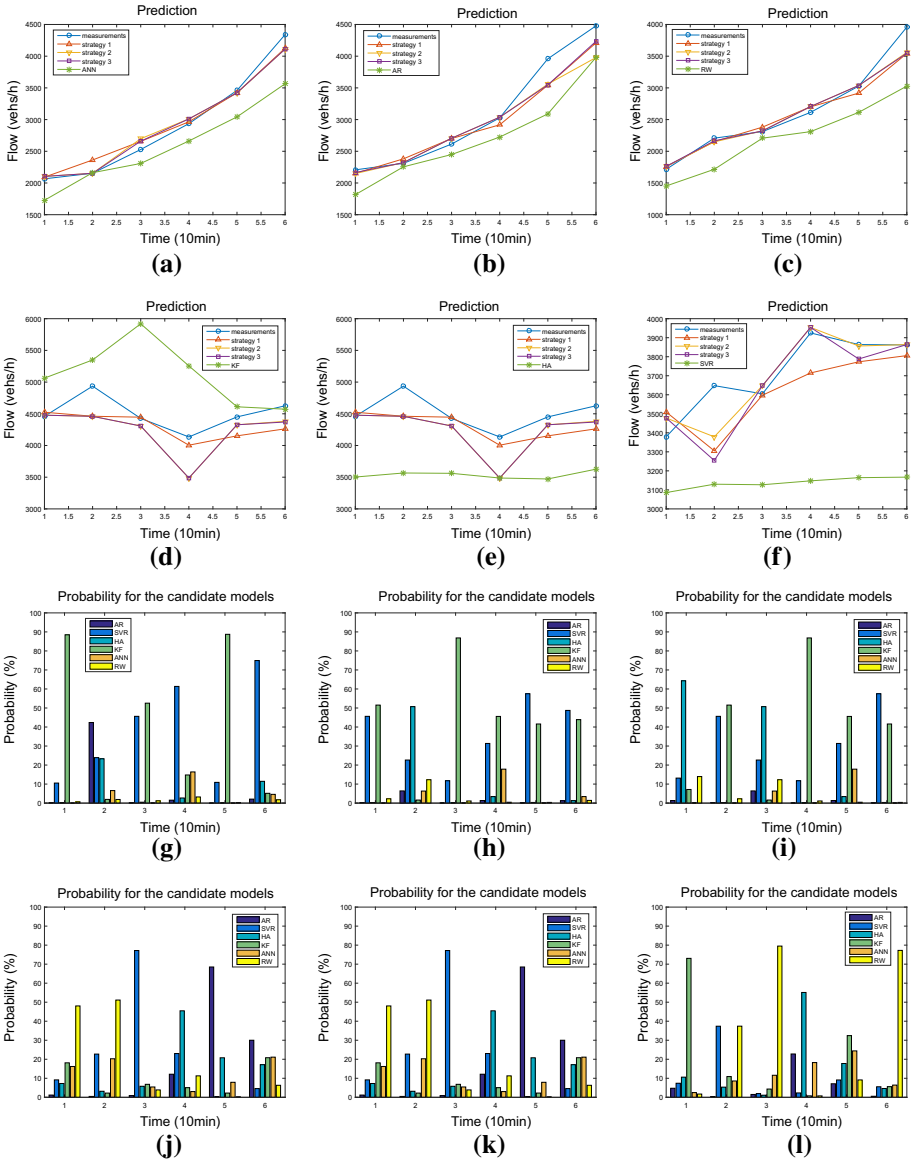


Fig. 6 **a–f** Predictions of various methods under several typical yet challenging scenarios and **g–l** the corresponding suitability probabilities

Figure 6a–c show the measurements and predictions of various methods for three scenarios, where the traffic flow quickly increases from around 2000 vehs/h to around 4000–4500 vehs/h in one hour at the racing time. In these cases, ANN, AR and RW are incapable of achieving accurate results, as shown in Fig. 6a–c. Figure 6g–i demonstrate the our framework assign these models lower suitability probabilities in these scenarios.

The underlying rationale is that ANN and AR models highly depend on the quality of the training data. They tend to *remember* the historical traffic flow, and when similar traffic flow

appears, they figure out the prediction based on what they remembered before. However, this mechanism usually fails in the cases where the traffic flow is dramatically changed. It is obvious that there are great gaps between the prediction results obtained from ANN and AR and the measurements, as shown in Fig. 6a, b, while our framework, regardless of which strategy is employed, can achieve satisfactory predictions close to the measurements. In Fig. 6c, the RW model also cannot deal with the large variations of traffic flow within a short period, leading to inaccurate predictions. However, the reason is theoretically different with that of ANN and AR models. The RW model always walks back one step, which means that it predicts the traffic flow at next moment as the current moment, so the predictions fail to reflect the sudden changes of traffic flow. It is worthwhile to note that, in all the three cases, our framework performs well to generate reliable forecasting results close to the ground truth.

Figure 6d, e show the measurements and predictions of various methods in a case that an accident caused traffic congestion. The traffic flow quickly drops from around 5000 vehs/h to around 4200 vehs/h within 20 min. While the KF model overshoots the ground truth by a large margin (as shown in Fig. 6d), the HA model cannot quickly response to this accident (as shown in Fig. 6e). The corresponding probabilities for Fig. 6d, e are shown Fig. 6j, k, respectively. In these two cases, the KF and HA are assigned with relatively low probabilities. For example, in Fig. 6d, the Kalman filter is overshooting from the 3rd 10-min to the 4th 10-min. Figure 6j shows the probability given by the deep network, in which the probability of the Kalman filter is relatively low for the 3rd 10-min to the 4th 10-min. Thus, no matter what strategy is chosen, the negative influence of the Kalman filter will be mitigated. Note that the forecasting result every step may not be as good as the most suitable candidate model every step, but our framework has learned to assigned the suitable candidate models with high probability. From Fig. 5, we can see that the suitable candidates are often assigned high probabilities every step, so the overall performance for a week or longer will be much better than that of individual candidates, since the performances of the individuals vary from time to time. It is why our framework achieves much better performance than the candidate models in a week time or longer.

Figure 6f shows the measurements and predictions of various methods in an afternoon where the traffic flow varies quite greatly. In such a case, SVR model's predictions deviated far from the measurements. This is because the SVR model is trained in a supervised manner to learn a function between the input traffic flow and the output by mapping the input into a higher dimensional space, and it is hard to train a very effective SVR model within limited training dataset. The bad performance of SVR model can also be observed in Fig. 5a–f. Note that our framework can still achieve good forecasting results in this case.

In the end, we compare the performance of the three proposed strategies and the stacked autoencoder proposed by Lv et al. [30]. The stacked autoencoder is trained in a layerwise greedy fashion on the dataset of Amsterdam motorways. The spatial and temporal correlations are inherently considered in the model. The deep architecture of the SAE is set to [120, 60, 30]. The scaling parameter for l_2 weight regularization penalty in Eq. 4 is set to 0.1. The sparsity is set to 0.03. The top of the SAE is connected to a logistic regression layer. The forecasting results in 10-min granularity are shown Table 5. From Table 5, we can find that the framework outperforms the SAE model, which appears to attribute the following reasons. First, the logistic regression layer attaches to the top of the SAE may not take the best effect for all the traffic conditions in spite of the ability of the SAE to learn the internal relationship inside the traffic data. On the other hand, our framework is able to choose the suitable predictor dynamically. Second, the SAE maintains the traffic state in the hidden units which implicitly storage the historical traffic state. The SAE with logistic regression layer may find many feasible solutions for the training set, which may take the risk to choose a poor one. The ensemble of heterogeneous predictors helps to reduce this risk [60].

Table 5 The forecasting results of the proposed framework and the SAE model on dataset of Amsterdam motorways

		A1	A2	A4	A8
SAE	RMSE	295.91	203.24	219.68	160.79
	MAPE	11.92	10.23	11.87	12.03
Strategy1	RMSE	**220.10**	**166.13**	**171.15**	**120.28**
	MAPE	8.09	7.23	7.46	7.53
Strategy2	RMSE	234.34	189.24	176.68	123.68
	MAPE	7.86	6.91	7.37	7.34
Strategy3	RMSE	224.26	178.77	171.43	120.37
	MAPE	**7.70**	**6.85**	**7.21**	**7.21**

Bold represents the lowest RMSE or MAPE value

5 Conclusion

In this paper, we propose a novel learning-based framework to improve the accuracy of dynamic traffic flow forecasting by integrating a set of representative models that are widely used in intelligent transportation systems nowadays. We employ stacked autoencoder (SAE) to select an optimal model or an optimal subset of models to predict traffic flow according to the current situation in a real-time manner. A model-driven mechanism is developed to automatically label the traffic flow data and then the labeled data can be applied to train the SAE. Three strategies are also developed in the proposed framework to flexibly and effectively integrate the predictions from different candidate models. Extensive experiments demonstrate that our framework can overcome the limitations of previous models in dealing with large and sudden variations of traffic flow, and achieve much better forecasting performance. Future investigations include evaluating the framework on more real traffic flow datasets and promoting its application in intelligent transportation systems.

Acknowledgements This work was supported partially by the National Natural Science Foundation of China (Nos. 61472145, 61772206, U1611461), in part by Special Fund of Science and Technology Research and Development on Application From Guangdong Province(SF-STRDA-GD No. 2016B010124011), in part by Guangdong High-level personnel of special support program (No. 2016TQ03X319) and the Guangdong Natural Science Foundation (No. 2017A030311027, No. 2016A030313047). The authors would like to thank Dr. Yubin Wang, from SIM Industries, Sassenheim, Netherlands, who provides the careful collected and preprocessed traffic flow data from the motorways of Amsterdam.

References

1. Agarwal M, Maze TH, Souleyrette R (2005) Impacts of weather on urban freeway traffic flow characteristics and facility capacity. Transp Res Symp Mid-Cont 20(5):1121–1134
2. Ahmed MS, Cook AR (1979) Analysis of freeway traffic time-series data by using Box–Jenkins techniques. 722, Transportation Research Record, Washington
3. Ahmed SA, Cook AR (1982) Discrete dynamic models for freeway incident detection systems. Transp Plan Technol 7(4):231–242
4. Akata Z, Perronnin F, Harchaoui Z, Schmid C (2014) Good practice in large-scale learning for image classification. IEEE Trans Pattern Anal Mach Intell 36(3):507–520
5. Barimani N, Kian AR, Moshiri B (2014) Real time adaptive non-linear estimator/predictor design for traffic systems with inadequate detectors. Intell Transp Syst IET 8(3):308–321
6. Basu S, Karki M, Ganguly S, DiBiano R, Mukhopadhyay S, Nemani R (2015) Learning sparse feature representations using probabilistic quadtrees and deep belief nets. In: European symposium on artificial neural networks, ESANN, pp 367–375
7. Bhavathrathan B, Patil GR (2013) Analysis of worst case stochastic link capacity degradation to aid assessment of transportation network reliability. Procedia Soc Behav Sci 104(Supplement C):507–515

8. Bohlin T (1976) Four cases of identification of changing systems. Math Sci Eng 126:441–518
9. Boto-Giralda D, Díaz-Pernas FJ, González-Ortega D, Díez-Higuera JF, Antón-Rodríguez M, Martínez-Zarzuela M, Torre-Díez I (2010) Wavelet-based denoising for traffic volume time series forecasting with self-organizing neural networks. Comput Aided Civ Infrastruct Eng 25(7):530–545
10. Chan KY, Dillon TS, Singh J, Chang E (2012) Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. IEEE Trans Intell Transp Syst 13(2):644–654
11. Comert G, Bezuglov A (2013) An online change-point-based model for traffic parameter prediction. IEEE Trans Intell Transp Syst 14(3):1360–1369
12. Davarynejad M, Wang Y, Vrancken J, van den Berg J (2011) Multi-phase time series models for motorway flow forecasting. In: 2011 14th international IEEE conference on intelligent transportation systems (ITSC). IEEE, pp 2033–2038
13. Davis GA, Nihan NL (1991) Nonparametric regression and short-term freeway traffic forecasting. J Transp Eng 117:178–188
14. Dou Q, Chen H, Yu L, Zhao L, Qin J, Wang D, Mok VC, Shi L, Heng PA (2016) Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks. IEEE Trans Med Imaging 35(5):1182–1195
15. Ghosh B, Basu B, O'Mahony M (2009) Multivariate short-term traffic flow forecasting using time-series analysis. IEEE Trans Intell Transp Syst 10(2):246–254
16. Guo J, Huang W, Williams BM (2014) Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. Transp Res Part C Emerg Technol 43:50–64
17. Hamed MM, Al-Masaeid HR, Said ZMB (1995) Short-term prediction of traffic volume in urban arterials. J Transp Eng 121(3):249–254
18. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
19. Hong WC, Pai PF, Yang SL, Theng R (2006) Highway traffic forecasting by support vector regression model with tabu search algorithms. In: International joint conference on neural networks, 2006, IJCNN'06. IEEE, pp 1617–1621
20. Hong WC, Dong Y, Zheng F, Lai CY (2011) Forecasting urban traffic flow by SVR with continuous ACO. Appl Math Model 35(3):1282–1291
21. Hu W, Yan L, Liu K, Wang H (2015) A short-term traffic flow forecasting method based on the hybrid PSO-SVR. Neural Process Lett 43:155–172
22. Huang W, Song G, Hong H, Xie K (2014) Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans Intell Transp Syst 15(5):2191–2201
23. Jeong YS, Byon YJ, Mendonca Castro-Neto M, Easa SM (2013) Supervised weighting-online learning algorithm for short-term traffic flow prediction. IEEE Trans Intell Transp Syst 14(4):1700–1707
24. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093
25. Kumar K, Jain V (1999) Autoregressive integrated moving averages (ARIMA) modelling of a traffic noise time series. Appl Acoust 58(3):283–294
26. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
27. Levin M, Tsao YD (1980) On forecasting freeway occupancies and volumes. J Transp Res Rec 773:47–49
28. Lippi M, Bertini M, Frasconi P (2013) Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning. IEEE Trans Intell Transp Syst 14(2):871–882
29. Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. Math Program 45(1–3):503–528
30. Lv Y, Duan Y, Kang W, Li Z, Wang FY (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16(2):865–873
31. Maria J, Amaro J, Falcao G, Alexandre LA (2016) Stacked autoencoders using low-power accelerated architectures for object recognition in autonomous systems. Neural Process Lett 43(2):445–458
32. Mathew TV, Rao KVK (2007) Introduction to transportation engineering. National Programme on Technology Enhanced Learning, Powai
33. Messer CJ (1993) Advanced freeway system ramp metering strategies for Texas. Technical report, Texas Transportation Institute, Texas
34. Moayedi HZ, Masnadi-Shirazi M (2008) ARIMA model for network traffic prediction and anomaly detection. In: International symposium on information technology, 2008. ITSim 2008, vol 4. IEEE, pp 1–6
35. Mori U, Mendiburu A, lvarez M, Lozano JA (2015) A review of travel time estimation and forecasting for advanced traveller information systems. Transportmetrica A Transp Sci 11(2):119–157. https://doi.org/10.1080/23249935.2014.932469

36. Okutani I, Stephanedes YJ (1984) Dynamic prediction of traffic volume through Kalman filtering theory. Transp Res Part B Methodol 18(1):1–11
37. Pan T, Sumalee A, Zhong RX, Indra-Payoong N (2013) Short-term traffic state prediction based on temporal-spatial correlation. IEEE Trans Intell Transp Syst 14(3):1242–1254
38. Peng Y, Lei M, Li JB, Peng XY (2014) A novel hybridization of echo state networks and multiplicative seasonal ARIMA model for mobile communication traffic series forecasting. Neural Comput Appl 24(3–4):883–890
39. Ross P (1982) Exponential filtering of traffic data. 869, Transportation Research Board, Washington
40. Singh K, Li B (2012) Estimation of traffic densities for multilane roadways using a Markov model approach. IEEE Trans Ind Electron 59(11):4369–4376
41. Smith BL, Demetsky MJ (1994) Short-term traffic flow prediction: neural network approach. J Transp Res Rec 1453:98–104
42. Smith BL, Demetsky MJ (1997) Traffic flow forecasting: comparison of modeling approaches. J Transp Eng 123(4):261–266
43. Srivastava N (2013) Improving neural networks with dropout. PhD thesis, University of Toronto
44. Stephanedes YJ, Michalopoulos PG, Plum RA (1981) Improved estimation of traffic flow for real-time control. J Transp Res Rec 795:28–39
45. Sumalee A, Zhong R, Pan T, Szeto W (2011) Stochastic cell transmission model (SCTM): a stochastic dynamic traffic model for traffic state surveillance and assignment. Transp Res Part B Methodol 45(3):507–533
46. Szeto MW, Gazis DC (1972) Application of Kalman filtering to the surveillance and control of traffic systems. Transp Sci 6(4):419–439
47. Tchrakian TT, Basu B, O'Mahony M (2012) Real-time traffic flow forecasting using spectral analysis. IEEE Trans Intell Transp Syst 13(2):519–526
48. Tomczak JM (2015) Learning informative features from restricted Boltzmann machines. Neural Process Lett pp 1–16
49. Tomczak JM, Gonczarek A (2016) Learning invariant features using subspace restricted Boltzmann machine. Neural Process Lett 45:173–182
50. Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning, ACM, pp 1096–1103
51. van Hinsbergen CP, Schreiter T, Zuurbier FS, Van Lint J, van Zuylen HJ (2012) Localized extended Kalman filter for scalable real-time traffic state estimation. IEEE Trans Intell Transp Syst 13(1):385–394
52. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 11:3371–3408
53. Vlahogianni EI, Karlaftis MG, Golias JC (2005) Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transp Res Part C Emerg Technol 13(3):211–234
54. Wang Y, van Schuppen JH, Vrancken J (2014) Prediction of traffic flow at the boundary of a motorway network. IEEE Trans Intell Transp Syst 15(1):214–227
55. Wang Y, Cheng JZ, Ni D, Lin M, Qin J, Luo X, Xu M, Xie X, Heng PA (2016) Towards personalized statistical deformable model and hybrid point matching for robust MR-TRUS registration. IEEE Trans Med Imaging 35(2):589–604
56. Williams BM, Hoel LA (2003) Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. J Transp Eng 129(6):664–672
57. Williams BM, Durvasula PK, Brown DE (1998) Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. Transp Res Rec J Transp Res Board 1644(1):132–141
58. Xie Y, Zhang Y, Ye Z (2007) Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. Comput Aided Civ Infrastruct Eng 22(5):326–334
59. Yuan Y, Van Lint J, Wilson RE, van Wageningen-Kessels F, Hoogendoorn SP (2012) Real-time Lagrangian traffic state estimator for freeways. IEEE Trans Intell Transp Syst 13(1):59–70
60. Zhou T, Han G, Xu X, Lin Z, Han C, Huang Y, Qin J (2017) δ-agree AdaBoost stacked autoencoder for short-term traffic flow forecasting. Neurocomputing 247:31–38. https://doi.org/10.1016/j.neucom.2017.03.049
61. Zhu JZ, Cao JX, Zhu Y (2014) Traffic volume forecasting based on radial basis function neural network with the consideration of traffic flows at the adjacent intersections. Transp Res Part C Emerg Technol 47:139–154