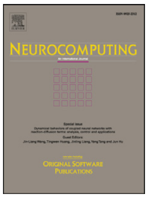
Neurocomputing 247 (2017) 31–38

Contents lists available at ScienceDirect

Neurocomputing

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



Brief papers

δ-agree AdaBoost stacked autoencoder for short-term traffic flow forecasting

Teng Zhou[a], Guoqiang Han[a], Xuemiao Xu[a,*], Zhizhe Lin[b], Chu Han[c], Yuchang Huang[d], Jing Qin[e]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong, 510006, China
[b] Affiliated Shantou Hospital of Sun Yat-sen University, Shantou, Guangdong, 515000, China
[c] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, 999077, China
[d] College of Mathematics and Information, South China Agricultural University, Guangzhou, Guangdong, 510642, China
[e] Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong, 999077, China

ARTICLE INFO

Article history:
Received 14 July 2016
Revised 10 February 2017
Accepted 19 March 2017
Available online 27 March 2017

Keywords:
Short-term traffic flow forecasting
AdaBoost
Stacked autoencoder
Time-series model
Dynamic system

ABSTRACT

Accurate and timely traffic flow forecasting is critical for the successful deployment of 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. Recently, the stacked autoencoder has been proven promising for traffic flow forecasting but still exists some drawbacks in certain conditions. In this paper, a training samples replication strategy is introduced to train a series of stacked autoencoders and an adaptive boosting scheme is proposed to ensemble the trained stacked autoencoders to improve the accuracy of traffic flow forecasting. Furthermore, sufficient experiments have been conducted to demonstrate the superior performance of the proposal.

1. Introduction

Traffic flow forecasting, especially short-term traffic flow forecasting, is a critical issue for intelligent transportation systems, because the traffic control actions highly depend on the accurate forecasting of traffic flow. Besides, the public can also benefit from the proactive forecasting.

Traffic flow does not only exhibit seasonality obscured by noise, but also reveals stochastic behaviors, which are affected by exogenous factors, such as unexpected incidents or weather [1]. Hence, this task is still a great challenge due to the large variation and inherent uncertainties of traffic flow.

A variety of theories and approaches have been proposed for traffic flow forecasting in the literature [2]. The conventional forecasting approaches can be generally classified into non-parametric methods and the parametric ones. Historical average [3], Kalman filtering methods [4–6], exponential smoothing [7–9], auto-regressive integrated moving average (ARIMA) model [10–12], seasonal autoregressive integrated moving average (SARIMA) [13–15] multivariate time series models [16–18], spectral analysis [1,19] and the structural time-series model [20,21] are grouped into parametric approaches, whereas artificial neural network (ANN) [22,23], non-parameter regression models [24], support vector machines (SVMs) [25,26], fuzzy logic system methods [27–29], and support vector regression (SVR) [30,31] are the non-parametric ones. However, the existing techniques have their drawbacks. For example, the historical average is paralyzed to the unexpected incidents; the Kalman filtering is prone to producing overshoot; the learning based methods are high depended on the quality of the training samples. Moreover, these methods require a considerable amount of engineering skill and domain expertise of the local traffic condition.

Recently, deep learning has drawn a lot of academic and industrial intentions, which can automatically discover the implicit relationships inside the data using a general-purpose learning procedure [32]. Deep learning techniques have also been proven promising for traffic flow forecasting [33,34]. Huang et. al. [33] and Lv et. al. [34] applied deep belief networks (DBN) and stacked autoencoder (SAE) to this task, respectively. They trained the deep architectures by minimizing the error between the outputs and the ground truth, which learn the inherently spatial and temporal correlations with rich accounts of data. Both of their methods

[*] Corresponding author.
E-mail address: xuemx@scut.edu.cn (X. Xu).

http://dx.doi.org/10.1016/j.neucom.2017.03.049
0925-2312/© 2017 Elsevier B.V. All rights reserved.

are demonstrated effective and efficient for real-time traffic flow forecasting.

The traffic flow forecasting by deep learning techniques involves sequential inputs of the traffic state to predict the traffic state next moment. The deep networks maintain the traffic state in the hidden units that implicitly storage the historical traffic state. However, the statistical problem arises when the historical traffic data is not enough, since the deep learning techniques may find many feasible predictors to give accuracy predictions for the training data, but fail to forecast the unknown cases [35]. Although the traffic flow data are easily collected by loop detectors, the traffic condition varies with the development of economy and society so rapidly that the period of training data cannot last too long. For small traffic flow data sets, unsupervised pre-training helps to prevent overfitting [36]. In a recent theoretical and empirical research, large numbers of saddle points with zero gradients are scattered on the landscape [37], especially for datasets with low dimensions such as traffic flow data. Around these points, the gradient is upward in most dimensions and downward in the remainder [38]. Thus, a individual network for traffic flow forecasting may stick in these poor local minima, i.e., the computational problem. Due to the large variation and inherent uncertainties of traffic flow, there are unknown cases that cannot be predicted by the trained network, because the true hypothesis may be outside the hyperspace supported by the training data set.

The most common way to deal with these shortcomings is to increase the size of the deep networks, such as the depth and the width [39]. Lots of recent publications focus on training deeper networks with a large amount of training data [40–42], and the computational resources are exponentially increasing with the increase of the network size. Meanwhile, some researchers attempt to ensemble relatively shallow networks to reduce the blow-up of uncontrolled computational complexity. For example, Cortes et. al. [43] reported achieving higher performance by DeepBoost algorithm on UCI datasets and MNIST datasets. The main idea behind this algorithm is drawn different weight to different deep hypotheses. Another idea by Huang et. al. [44] is to build a boosting model according to the reconstruction error of the training data, whose idea is that the result becomes less reliable when the reconstruction error increases. These attempts somewhat widen the networks by adaptive projections in the top layer, as opposite to the fixed nonlinear projections, such as sigmoid or ReLU.

In this study, we introduce a boosting scheme for the stacked autoencoder network to improve the accuracy of traffic flow forecasting. Comparing with [43,44], there are two purposes of our boosting scheme. The first one is similar but not exactly the same as [44], we use prediction error to retrain the stacked autoencoders by rearranging the training data, since the reconstruction error is the measurement of the ability of the deep network to reproduce the input, while the prediction error is the measurement of the generalization ability of the deep network. The second idea comes from the weather forecasting [45,46] and ocean modeling [47]. In the climate forecasting, a series of simulation models run under variant initial settings to forecast a series possible projections of future weather. The weighted average serves as the best guess of the future weather, since individual model biases may be partially canceled. In practice, the weighted average is likely to be more accuracy than any individual prediction [46]. Different from the climate forecasting tasks, we use an adaptive ensemble strategy to integrate the stacked autoencoders, which has been proven to be a useful tool to handle uncertainties in model initial conditions, model parameters, and model structures [48,49]. In our framework, the training data are separated for training and for validation. The prediction error on the validation set is calculated. Then the subsequent SAE will get more simulations by rearranging the training data according to this error. The importance of this SAE is determined by the prediction error. Finally, we exhaustively search over all feasible traffic flow rate to find a prediction to satisfy as many as possible predictions by the ensemble SAEs according to their importance.

The contributions of this paper can be summarized as follows:

- A training sample replication strategy is introduced to train a series of stacked autoencoders;
- An adaptive boosting scheme is proposed to ensemble the trained stacked autoencoders;
- Sufficient experiments are conducted to demonstrate the performance of our proposal.

The rest of this paper is organized as follows. The second part is the methodology and the third is the empirical study of the real world data from Amsterdam, Netherland. Then the conclusions follow.

2. Methodology

In this section, the stacked autoencoder (SAE) deep learning technique is employed to construct multiple models for traffic flow forecasting. And then an ensemble scheme based on δ-agree AdaBoost regression is introduced to boost the learned models.

2.1. Stacked autoencoder

The stacked autoencoder network is one of state-of-the-art deep learning techniques. SAE is a kind of neural networks, whose layers are unsupervised trained layer-wise. Each layer is trained by constructing an autoencoder. An autoencoder is a neural network with only one input layer, one hidden layer, and one output layer. The output layer is expected to reproduce the input, so the hidden layer can be seen as a kind of encodings of the input layer. After the layer-wise training, the weights of the layers could be initialized to sensible local suboptimal [32]. Then the SAE will achieve a discriminant one by fine-tuning with the labeled data. See [34] for more details.

Unlike the deep convolutional neural network (DCNN), the SAE is a full-connected network. One of the motivations of the convolutional and pooling layers of DCNN is to reduce the spatial and temporal complexity by ultimately moving the fully connected architectures to sparsely connected ones [39], as it can hardly learn a deep full-connected network from the full-size image data sets due to the dramatically computational complexity. The dimension of traffic flow data are relatively low comparing to that of the images, so it is reasonable to construct full-connected networks. Lv et. al. [34] have demonstrated that the SAE network can successfully discover the spatial and temporal correlations from the traffic flow data. Hence, the SAE [34] is taken as the preliminary model to boost in this study.

2.2. δ-agree AdaBoost regression

Although the SAE is demonstrated promising, robust and comparable in the reported study [34], the SAE for traffic flow forecasting may suffer from some drawbacks in certain conditions. As analyzed, the statistical problem occurs when the amount of data for training is small as the input period of the data would not last too long. There may be a large amount of feasible solutions for the training data, some of which may have poor generalization abilities. If only one individual SAE is employed for traffic flow forecasting, we are at the risk of choosing a poor one. This encourages us to construct an ensemble of a series of SAEs, whose votes may help to reduce the risk of choosing a poor prediction. The computational problem is inevitable, since the optimal training of neural networks is NP-hard [50]. Although the poor local minima are not a serious problem that the networks often reach similar generalization performance in practice, the greedy nature of gradient descent optimization still pushes us to the edge of this danger. An ensemble constructed by different local search with different deep architectures or different initial value may have a better approximation than the individual preliminary. This motivates us to ensemble SAEs with different architectures and different initial value, and rearrange the training data to retrain the subsequence SAEs. The representational problem is subtle for the traffic flow forecasting tasks. Since the training data are finite, the deep learning algorithms will stop searching when they fit all the training samples. The groundtruth hypotheses may be outside the hyperspace supported by the training samples. The weighted combination of the hypotheses is able to expand the hyperspace of training samples.

The deep architectures for the SAEs are explored in [33,34]. Both of them report that the feasible number of the hidden layers is around three and the feasible number of hidden units is no more than 500. For the consideration of computational complexity, the candidate deep architectures of this study range from one to four hidden layers with 10–300 hidden units following the suggestion in [34].

The aforementioned stacked autoencoder aims to learn the hypothesis denoted as $\mathcal{G}(\cdot)$, which predicts the future traffic flow rate by given the current traffic flow rate. To further depict the ensemble, we firstly define each group of traffic flow data x as $\{v_{i,j}\}_{i=1,\dots,M,j=t,\dots,t-N+1}$, where $v_{i,\,j}$ is the traffic flow data on the ith measurement location at jth time interval. The traffic flow prediction of next time interval on the ith location by kth SAE network can be denoted as $\mathcal{G}_i^{(k)}(x)$. Without loss generality, we simplify the hypothesis $\mathcal{G}_i^{(k)}(x)$ as $\mathcal{G}^{(k)}(x)$ omitting the location indicator i. And we also denote the prediction of the sth sample as $\hat{y}_s = \mathcal{G}^{(k)}(x_s)$. Then the training data set can be denoted as $\mathcal{T} = \{(x_s, y_s)\}_{s=1,\dots,S}$, where y_s is the groundtruth of $\hat{y}_s$, and S is number of training samples. The boosting algorithm discussed following is always focused on a certain location, and it is easy to be extended to all locations.

In order to improve the forecasting accuracy, we encourage the SAE to get more stimulations by training samples with large prediction error, which often occurs when the traffic flow is heavy. Actually, these moments are critically important to the intelligent transportation system, which are likely to be the commuting time and easily congested. We introduce a δ-agree scheme for the boosting phase. The δ-agree scheme is defined as a discriminative function in Eq. (1).

$$\mathcal{I}(|\mathcal{G}^{(k)}(x_s) - y_s| - \delta), \tag{1}$$

where $\mathcal{I}(x) = \begin{cases} 1, & \text{if} x > 0, \\ -1, & \text{otherwise}. \end{cases}$

Eq. (1) means the case that the prediction error exceeds δ will take positive effect in the succeeding weighting scheme, vice versa. This parameter separates the forecasting results into two part as a latent parameter. For some extreme cases, the prediction error is large. Then the next SAE will be trained by taking more consideration on these cases.

Then we introduce a weight $w_s^{(k)}$ for every sample for the kth SAE. Initially, the weight of the samples is equal $w_s^{(1)} = \frac{1}{S}$ for the first SAE. The discriminative error of the SAE is calculated as:

$$\varepsilon^{(k)} = \frac{1}{2}\sum_{s=1}^{S} w_s^{(k)}[\mathcal{I}(|\mathcal{G}^{(k)}(x_s) - y_s| - \delta) + 1]. \tag{2}$$

Then the importance of this SAE is determined by its discriminative error as:

$$\alpha^{(k)} = \frac{1}{2}\log\frac{1-\varepsilon^{(k)}}{\varepsilon^{(k)}}. \tag{3}$$

In Eq. (3), the smaller discriminative error of the SAE achieves, the more importance it gains. The new weights of the samples can be updated according to the discriminative error and the importance of this SAE.

$$w_s^{(k+1)} = \frac{w_s^{(k)}}{\mathcal{Z}^{(k)}} e^{\alpha^{(k)}\mathcal{I}(|\mathcal{G}^{(k)}(x_s)-y_s|-\delta)}, \tag{4}$$

where $\mathcal{Z}^{(k)} = \sum_{s=1}^{S} w_s^{(k)} e^{\alpha^{(k)}\mathcal{I}(|\mathcal{G}^{(k)}(x_s)-y_s|-\delta)}$ is a normalization factor.

In order to let the next SAE get more stimulations with these extreme cases of large prediction error, so we expand the training data set by introducing a replication factor $r_s^{(k)} = Cw_s^{(k)}S$. C is a constant indicating the average replication times, in our experiment C is set to 100.

We replicate the sth training sample $r_s^{(k)}$ times (rounding-off) to construct a new data set. With this data set, we try to train the SAE with different deep architectures and initial values. The best deep architecture for this data set is by cross-validation of the candidate architectures.

After all the SAEs are trained, for a testing sample of traffic flow x, the prediction is depicted as:

$$\hat{y} = argmin_{\hat{y}\in[0,v_{max}]} \sum_{k=1}^{K} \alpha^{(k)}\mathcal{I}(|\mathcal{G}^{(k)}(x) - \hat{y}| - \delta), \tag{5}$$

where v_{max} is the maximum capacity of traffic flow rate on that location, and $\hat{y}$ is the traffic flow rate to predict, which is assumed as a natural number.

Firstly, K trained models are employed to make K predictions. Then we exhaustively enumerate all feasible traffic flow rate from 0 to v_{max} to search an optimal $\hat{y}$. In Eq. (5), $\alpha^{(k)}$ is the importance of the kth SAE, which is determined by Eq. (3) according to the discriminative error of the kth SAE. The larger discriminative error is, the less importance the kth SAE gains. On the other hand, $\mathcal{I}(\cdot)$ is -1 if the error between $\hat{y}$ and the prediction by the SAE is no more than δ. Thus, in order to minimize Eq. (5), the optimal $\hat{y}$ is expected to meet the predictions made by as many SAEs of high importance as possible. In another word, this value by Eq. (5) satisfies as many as possible predictions of the ensemble SAEs to eliminate the short-sight of the individual SAE according to their importance.

This algorithm can be summarized as follows.

Algorithm 1 Training the boosting algorithm for SAEs.

```
Require: $\mathcal{T} = \{(x_s, y_s)\}_{s=1,\dots,S}$, $\delta$, $C$
Ensure: $\alpha^{(k)}$, $\mathcal{G}^{(k)}(\cdot)$, $k = 1, \dots, K$
$k = 1$
$w_s^{(k)} = \frac{1}{S}$
while $k \le K$ do
    replicate $\mathcal{T}$ according to $r_s^{(k)} = Cw_s^{(k)}S$
    train and cross validate to choose the best deep architecture
      for the SAE $\mathcal{G}^{(k)}(\cdot)$ with the replicated samples
    calculate the discriminative error $\varepsilon^{(k)} =$
      $\frac{1}{2}\sum_{s=1}^{S} w_s^{(k)}[\mathcal{I}(|\mathcal{G}^{(k)}(x_s) - y_s| - \delta) + 1]$
    if $\varepsilon^{(k)} \ge \frac{1}{2}$ then
        continue
    end if
    calculate $\alpha^{(k)} = \frac{1}{2}\log\frac{1-\varepsilon^{(k)}}{\varepsilon^{(k)}}$
    update the weight $w_s^{(k+1)} = \frac{w_s^{(k)}}{\mathcal{Z}^{(k)}} e^{\alpha^{(k)}\mathcal{I}(|\mathcal{G}^{(k)}(x_s)-y_s|-\delta)}$
    $k = k + 1$
end while
```

3. Case study

In this section, the traffic flow data from four motorways A1, A2, A4 and A8 ending on the ring road (A10 motorway) of Amsterdam are used for the empirical study.

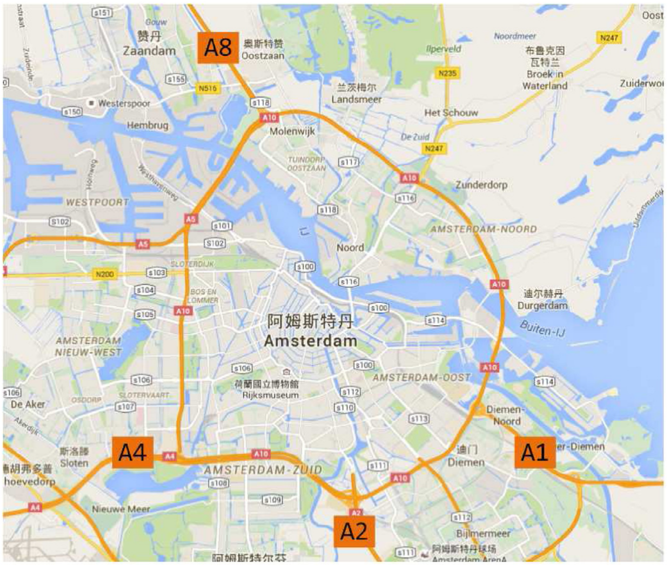


Fig. 1. The four motorways namely A1, A2, A4, and A8, which end on the ring road of Amsterdam.

3.1. Data description

The real world data was collected from four motorways by Wang et al. [51], namely A1, A2, A4, and A8, which end on the ring road of Amsterdam (the A10 motorway), as shown in Fig. 1. We simply depict the four motorways as follows. The four measurement sites locate on the motorways a short distance before the merge points to the ring road. The data was provided from May 20, 2010, to June 24, 2010, with 1-min aggregation, i.e., the number of vehicle per hour in a certain minute, collected by MONICA sensor.

- 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 interchange Hoevelaken in 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 raw data mix with incorrect measurements, which are zeros for a long period or negative values. We simply fill the incorrect data by averaging measurements of the same moments of other weeks.

3.2. 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. (6) and (7), respectively:

$$RMSE = \sqrt{\frac{1}{M}\sum_{m=1}^{M}(\hat{v}(m) - v(m))^2} \tag{6}$$

$$MAPE = \frac{1}{M}\sum_{m=1}^{M}\left|\frac{\hat{v}(m) - v(m)}{v(m)}\right| \times 100\% \tag{7}$$

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

3.3. Experimental setup

As discussed in [4,51,52], the study of traffic flow forecasting should not be of interest to predict minute-by-minute fluctuations. Therefore, the 10-min average, which is the average of 1-min aggregation in subsequent 10 min, is chosen for the forecasting task the same as [51].

The collected data are divided into two parts, the first four weeks are used for training and the rest are used for testing. The training data are divided into ten parts, nine of ten are for training the SAE, and the other is for validation after every training epoch. Following the instruction by Lv et. al. [34], the candidate deep architectures for the traffic forecasting tasks are limited to no more than 4 hidden layers, the hidden units of each layer are limited to no more than 300. The scaling parameter for weight regularization penalty in is set to 0.1. The sparsity is set to 0.03, see [34] for the interpretations of regularization penalty and sparsity. We randomly drop out some measurements of the training inputs to improve the network performance (see [53] for more details). The batch size in this tasks is the entire training samples, since the dimension of each sample is relatively small comparing to that of images. Thus, all the training samples pass forward and backward every iteration in one epoch without considering the limitations of the memory. The maximum iterations are limited to 10 k. Similarly, all the validation samples pass forward and backward every iteration in one epoch. The validation procedure is conducted after every training epoch. The optimization method is the limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) algorithm [54], a typical gradient descent algorithm.

Another two parameters are the number of ensembles K and the δ. To evaluate how these two parameters influence the forecasting performance, we conduct the experiments. The number of ensembles K is tested from 10 to 100. The forecasting performance at A1 is illustrated in Fig. 2 with a different number of ensembles. The RMSE decreases sharply when the number of ensemble K increases until 40, while the MAPE keeps decreasing until the number of ensemble reaches 70. The performance of the SAEs is boosted up as the number of ensembles increases in a certain range. However, if too many ensembles are integrated, the computation demand would be too large. The δ is a latent parameter to be set to separate the extreme cases of large prediction error. As defined in Eq. (1), if the error between the prediction and the groundtruth larger than δ, $\mathcal{I}$ will be 1, otherwise -1. Thus, the cases with large prediction error will weight more in Eq. (4). The δ is tested from 100 to 250 shown in Fig. 3. There may be two main reasons that the δ has a good level of tolerance. First, if some extreme cases with large prediction error appear in the prior SAE, the subsequent SAE is trained by taking more consideration on these cases. Then the subsequence SAE may well deal with these cases. Second, the ensemble SAEs with large discriminative error weight less in Eq. (3). These two parameters are listed in Table 1 for the succeeding experiments.

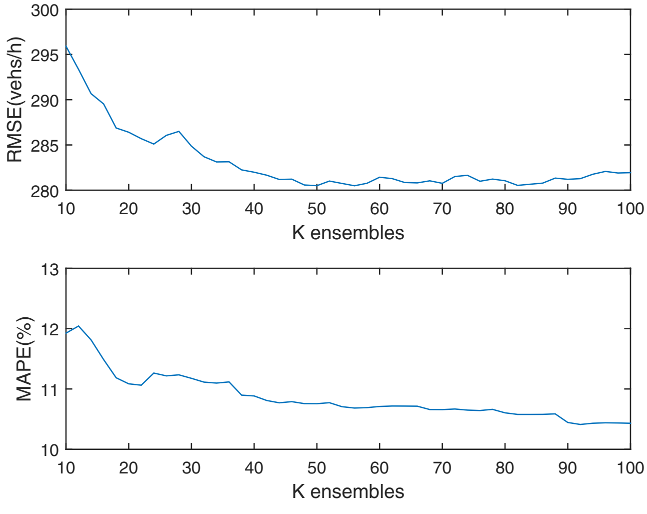


Fig. 2. Forecasting performance with different number of ensembles.

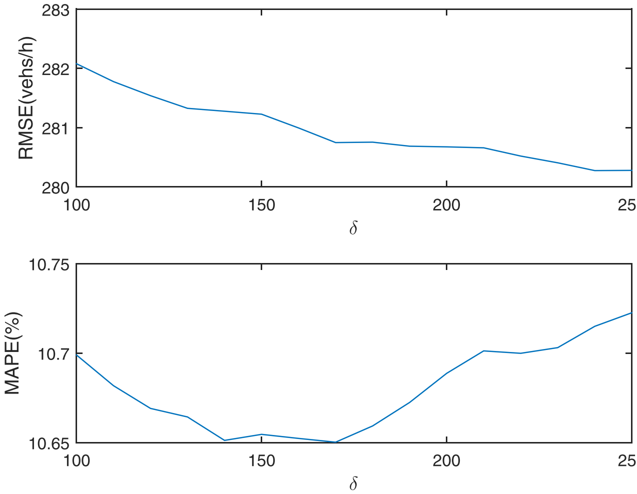


Fig. 3. Forecasting performance with different δ.

Table 1
The parameters K and δ for the succeeding experiments.

	A1	A2	A4	A8
K	70	50	90	70
δ	170	120	150	100

3.4. Performance evaluation

Table 3 have several comparisons with various commonly used models integrated in intelligent transportation systems including the state-of-the-art ones. The hybrid particle swarm optimization support vector regression method (SVR) is detailed in [55]. The historical average model (HA) and the random walk method (RW) are used as the control methods in [31]. These two methods are often used as the baseline for a new one. The artificial neural network (ANN) is detailed in [56]. The Kalman filtering method (KF) is detailed in [4]. The least squares boosting is the ensemble of simple models [57]. The stacked autoencoder method (SAE) is proposed by Lv et. al. [34]. The last one is our proposed method. The concepts of these control models are simply introduced as follows.

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

Table 2
Configurations of ANN model.

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

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

Auto regression. 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. 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.

Support vector machine regression. For the support vector machine regression model, several parameters need to be set beforehand. The regression horizon is set the same as AR model. We use radial basis function (RBF) as the kernel type in this study. The cost parameter C is set to the maximum difference between the traffic flow. The width parameter γ and the ϵ-insensitive are determined by particle swarm optimization. The width parameter γ for the RBF kernel is 3×10^{-6} and the ϵ-insensitive loss for the SVR is 1 in this study.

Kalman filtering. A wavelet denoising procedure proposed by Xie et al. [4] is employed to preprocess the traffic flow data. We use Daubechies 4 as the mother wavelet as suggesting in [4]. The variance of the process error Q is set as $0.1 \times \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The variance of the measurement noise is 0, since we regard the measurement is correct. The initial state is set to $\left[\frac{1}{n}, \ldots, \frac{1}{n}\right]$, where n is set to 8, the same as Xie et al. [4]. The covariance matrix of initial state estimation error is $10^{-2} \times \mathbf{I}$.

Artificial neural network. We employ the artificial neural networks introduced in Zhu et al. [56]. The network parameters are described in Table 2, where most of them are consistent with [56].

Least squares boosting. The least square boosting (LSBoost) is one of most popular boosting algorithm that ensembles linear regression. Kkdeniz et al. [57] have applied this method to load forecasting in energy day-ahead market And they declared that least squares boosting algorithm give more robust results than SARIMA method for load forecasting. In this experiment, the number of ensembles of LSBoost is set the same as that listed in Table 1.

Stacked autoencoder. The stacked autoencoder is trained in a layerwise greedy fashion. The spatial and temporal correlations are inherently considered in the model. The deep architecture of the SAE is set to [120, 60, 30] by cross-validation.

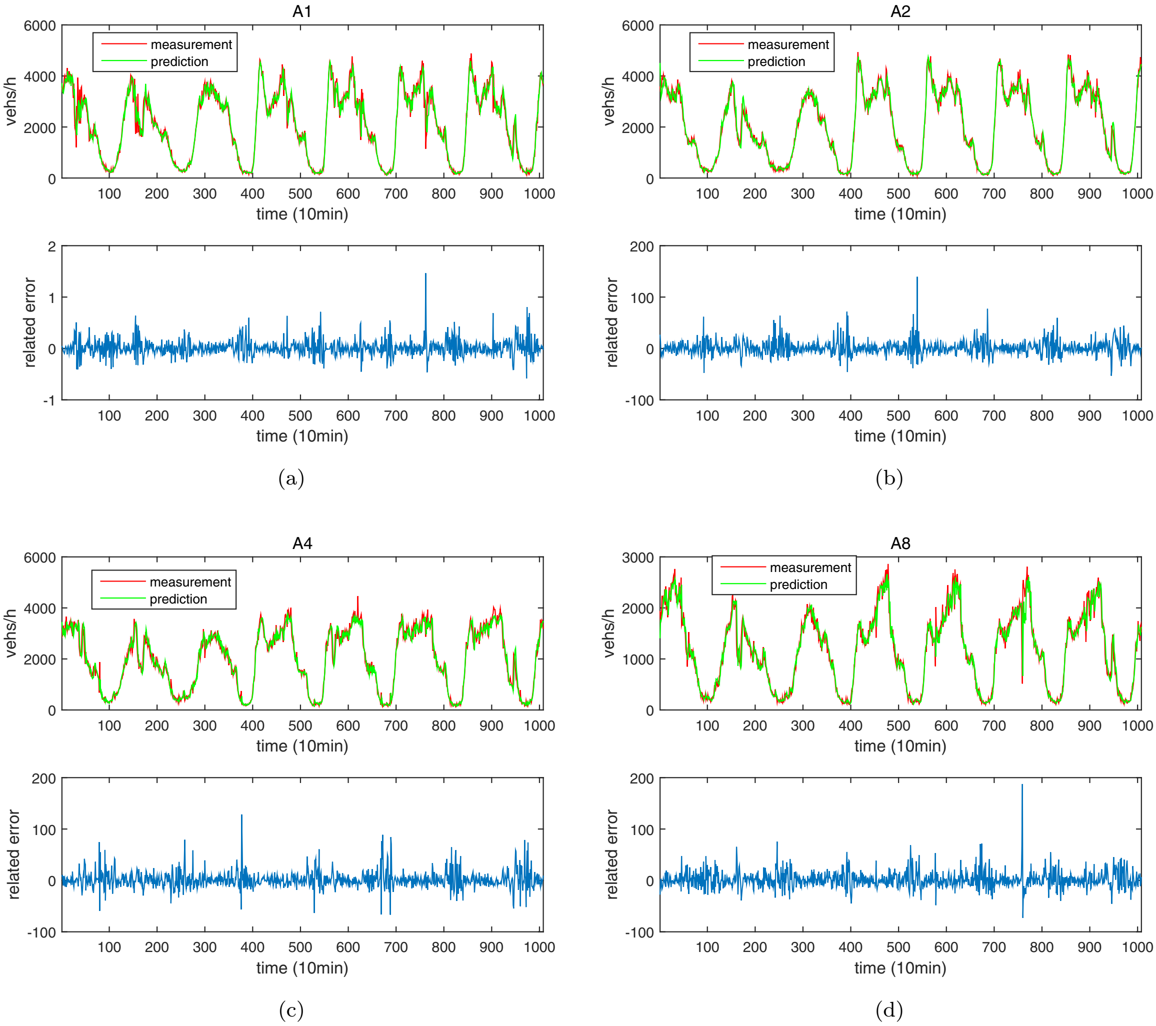
We compare the forecasting results of the proposed boosting method with the control models mentioned, which are often used in intelligent traffic systems. As illustrated in Table 3, the proposal achieves more accurate results than the controls. For example, comparing with the RMSEs of SAE, which achieves the best results among the others, the RMSEs of our proposal decrease 5.12%, 2.99%, 5.43%, and 2.95% at A1, A2, A4 and A8, respectively.

Table 3
The forecasting results of the proposed framework and the control models on dataset of Amsterdam motorways.

		A1	A2	A4	A8
SVR	RMSE	329.09	259.74	253.66	190.30
	MAPE	14.34	12.22	12.23	12.48
HA	RMSE	404.84	348.96	357.85	218.72
	MAPE	16.87	15.53	16.72	16.24
RW	RMSE	312.92	223.82	230.01	174.14
	MAPE	12.65	11.43	12.06	12.37
AR	RMSE	301.44	214.22	226.12	166.71
	MAPE	13.57	11.59	12.70	12.71
ANN	RMSE	299.64	212.95	225.86	166.50
	MAPE	12.61	10.89	12.49	12.53
KF	RMSE	332.03	239.87	250.51	187.48
	MAPE	12.46	10.72	12.62	12.63
LSBOOST	RMSE	306.33	233.88	229.78	177.52
	MAPE	13.78	14.43	12.90	14.00
SAE	RMSE	295.91	203.24	219.68	160.79
	MAPE	11.92	10.23	11.87	12.03
PROPOSAL	RMSE	280.75	197.16	207.75	156.04
	MAPE	10.65	9.85	11.06	11.63

The traffic flow may vary so largely at different moments a day, or different days, i.e., the training data may not cover all the cases occurring in the future, that the SAE is prone to the most similar results that it has learned. By taking consideration of the SAEs with different importance, the boosting procedure is able to ensemble the predictions of the SAEs to eliminate the short-sight of an individual one. Moreover, the individual SAE may be stuck in some poor local minima. A series of integrated SAEs of our method search from different initial values and directions are more easily to find a better prediction. We also contrast the results of the LSBoost, which is a boosting algorithm integrated simple models, i.e., linear regression, and the proposal outperforms the LSBoost. In addition, our method gets higher accuracy than the other control models, because our method inherits the advantages of the SAE, which can automatically discover the implicit relationships inside the data.

Finally, we report some forecasting scenarios to demonstrate the effectiveness of the proposed framework in dealing with variations and uncertainties of traffic flow. The predictions of the proposed method are drawn with a red line, while the measurements are done with a green line in Fig. 4. The related error drawn in blue line is the error between the measurements and the predictions divided by the measurements. As shown in Fig. 4, the proposal achieves relatively high accuracy most of the time, except the traffic flow is very low early in the morning or late at night. For these cases, small prediction error still causes a large related error. Fortunately, we are more likely to care about the forecasting accuracy when the traffic is really heavy.



(a)

(b)

(c)

(d)

Fig. 4. Fig. 4a–d are the predictions of proposal and the measurements in a week, and the prediction related error (the difference between the measurements and the predictions divided by the measurements), respectively. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

4. Conclusion

In this paper, we propose a novel ensemble method to improve the accuracy traffic flow forecasting by integrating stacked autoencoders that have been proven promising for traffic flow forecasting, but still suffer from some shortcomings in certain conditions. In order to eliminate the short-sight of an individual stacked autoencoder, we propose a boosting-up scheme to improve the forecasting accuracy. In this scheme, a training samples replication strategy is introduced to train a series of stacked autoencoders and a boosting algorithm is proposed to ensemble the trained SAEs. Extensive experiments demonstrate the proposal outperforming the stacked autoencoder in dealing with traffic flow forecasting, and achieving better forecasting performance. Future investigations include evaluating the method on more real traffic flow datasets and promoting its applications in intelligent transportation systems.

Acknowledgment

This work was supported partially by the National Natural Science Foundation of China under Grants No. 61300137, 61472145, in part by Special Fund of Science and Technology Research and Development on Application From Guangdong Province (Grant No. 2016B010124011). 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] Y. Zhang, Y. Zhang, A. Haghani, A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model, Transp. Res. Part C: Emerg. Technol. 43 (2014) 65–78.
[2] U. Mori, A. Mendiburu, M. Álvarez, J.A. Lozano, A review of travel time estimation and forecasting for advanced traveller information systems, Transportmetrica A: Transp. Sci. 11 (2) (2015) 119–157.
[3] Y.J. Stephanedes, P.G. Michalopoulos, R.A. Plum, Improved estimation of traffic flow for real-time control (discussion and closure), Transp. Res. Rec. (795) (1981) 28–39.
[4] Y. Xie, Y. Zhang, Z. Ye, Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition, Comput. Aided Civil Infrastruct. Eng. 22 (5) (2007) 326–334.
[5] J. Guo, W. Huang, B.M. Williams, Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification, Transp. Res. Part C: Emerg. Technol. 43 (2014) 50–64.
[6] N. Barimani, A.R. Kian, B. Moshiri, Real time adaptive non-linear estimator/predictor design for traffic systems with inadequate detectors, Intell. Transp. Syst. IET 8 (3) (2014) 308–321.
[7] P. Ross, Exponential filtering of traffic data, Transp. Res. Rec. 869 (1982) 43–49.
[8] C.J. Messer, Advanced Freeway System Ramp Metering Strategies for Texas, Technical Report, Texas Transportation Institute, 1993.
[9] K.Y. Chan, T.S. Dillon, J. Singh, E. Chang, 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) (2012) 644–654.
[10] H. Zare Moayedi, M. Masnadi-Shirazi, Arima model for network traffic prediction and anomaly detection, in: Proceedings of the International Symposium on Information Technology, 4, IEEE, 2008, pp. 1–6.
[11] G. Comert, A. Bezuglov, An online change-point-based model for traffic parameter prediction, IEEE Trans. Intell. Transp. Syst. 14 (3) (2013) 1360–1369.
[12] Y. Peng, M. Lei, J.-B. Li, X.-Y. Peng, A novel hybridization of echo state networks and multiplicative seasonal Arima model for mobile communication traffic series forecasting, Neural Comput. Appl. 24 (3–4) (2014) 883–890.
[13] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal Arima process: theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.
[14] J. Guo, B.M. Williams, B.L. Smith, Data collection time intervals for stochastic short-term traffic flow forecasting, Transp. Res. Rec. J. Transp. Res. Board 2024 (1) (2008) 18–26.
[15] G. Shi, J. Guo, W. Huang, B.M. Williams, Modeling seasonal heteroscedasticity in vehicular traffic condition series using a seasonal adjustment approach, J. Transp. Eng. 140 (5) (2014).
[16] Y. Kamarianakis, P. Prastacos, Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches, Transp. Res. Rec. J. Transp. Res. Board 1857 (1) (2003) 74–84.
[17] W. Min, L. Wynter, Real-time road traffic prediction with spatio-temporal correlations, Transp. Res. Part C: Emerg. Technol. 19 (4) (2011) 606–616.
[18] T. Ma, Z. Zhou, B. Abdulhai, Nonlinear multivariate time–space threshold vector error correction model for short term traffic state prediction, Transp. Res. Part B: Methodol. 76 (2015) 27–47.
[19] T.T. Tchrakian, B. Basu, M. O'Mahony, Real-time traffic flow forecasting using spectral analysis, IEEE Trans. Intell. Transp. Syst. 13 (2) (2012) 519–526.
[20] J. Durbin, S.J. Koopman, Time Series Analysis by State Space Methods, Oxford University Press, 2012. 38
[21] B. Ghosh, B. Basu, M. O'Mahony, Multivariate short-term traffic flow forecasting using time-series analysis, IEE Trans. Intell. Transp. Syst. 10 (2) (2009) 246–254.
[22] H. Liu, H. van Zuylen, H. van Lint, M. Salomons, Predicting urban arterial travel time with state-space neural networks and Kalman filters, Transp. Res. Rec. J. Transp. Res. Board 1968 (1) (2006) 99–108.
[23] H.T. Siegelmann, E.D. Sontag, Turing computability with neural nets, Appl. Math. Lett. 4 (6) (1991) 77–80.
[24] G.A. Davis, N.L. Nihan, Nonparametric regression and short-term freeway traffic forecasting, J. Transp. Eng. 117 (1991) 178–188.
[25] M. Davarynejad, Y. Wang, J. Vrancken, J. van den Berg, Multi-phase time series models for motorway flow forecasting, in: Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), IEEE, 2011, pp. 2033–2038.
[26] W.-C. Hong, Y. Dong, F. Zheng, C.-Y. Lai, Forecasting urban traffic flow by SVR with continuous ACO, Appl. Math. Model. 35 (3) (2011) 1282–1291.
[27] Y. Zhang, Z. Ye, Short-term traffic flow forecasting using fuzzy logic system methods, J. Intell. Transp. Syst. 12 (3) (2008) 102–112.
[28] O.A. Arqub, Z. Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415.
[29] O.A. Arqub, M. Al-Smadi, S. Momani, T. Hayat, Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems, Soft Comput. (2016) 1–16.
[30] W.-C. Hong, P.-F. Pai, S.-L. Yang, R. Theng, Highway traffic forecasting by support vector regression model with tabu search algorithms, in: Proceedings of the International Joint Conference on Neural Networks, IEEE, 2006, pp. 1617–1621.
[31] M. Lippi, M. Bertini, P. Frasconi, Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning, IEE Trans. Intell. Transp. Syst. 14 (2) (2013) 871–882.
[32] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[33] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.
[34] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Trans. Intell. Transp. Syst. 16 (2) (2015) 865–873.
[35] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2016) 11–26.
[36] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[37] Y.N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2933–2941.
[38] A. Choromanska, M. Henaff, M. Mathieu, G.B. Arous, Y. LeCun, The loss surfaces of multilayer networks., in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2015.
[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[40] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[41] V.N. Murthy, V. Singh, T. Chen, R. Manmatha, D. Comaniciu, Deep decision network for multi-class image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[42] F.N. Iandola, M.W. Moskewicz, K. Ashraf, K. Keutzer, Firecaffe: near-linear acceleration of deep neural network training on compute clusters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[43] C. Cortes, M. Mohri, U. Syed, Deep boosting, in: Proceedings of the 31st International Conference on Machine Learning (ICML-14), 2014, pp. 1179–1187.
[44] W. Huang, N. Zhang, W. Hu, H. Hong, G. Song, K. Xie, Dynamic boosting in deep learning using reconstruction error, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 473–480.
[45] M. Thomson, F. Doblas-Reyes, S. Mason, R. Hagedorn, S. Connor, T. Phindela, A. Morse, T. Palmer, Malaria early warnings based on seasonal climate forecasts from multi-model ensembles, Nature 439 (7076) (2006) 576–579.
[46] R.E. Chandler, Exploiting strength, discounting weakness: combining information from multiple climate simulators, Philos. Trans. R. Soc. Lond. A: Math. Phys. Eng. Sci. 371 (1991) (2013) 20120388.
[47] F. Doblas-Reyes, V. Pavan, D. Stephenson, The skill of multi-model seasonal forecasts of the wintertime north atlantic oscillation, Clim. Dyn. 21 (5–6) (2003) 501–514.

[48] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 993–1001.

[49] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, CRC Press, 2012.

[50] A.L. Blum, R.L. Rivest, Training a 3-node neural network is np-complete, Neural Netw. 5 (1) (1992) 117–127.

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

[52] D. Boto-Giralda, F.J. Díaz-Pernas, D. González-Ortega, J.F. Díez-Higuera, M. Antón-Rodríguez, M. Martínez-Zarzuela, I. Torre-Díez, Wavelet-based denoising for traffic volume time series forecasting with self-organizing neural networks, Comput. Aided Civil Infrastruct. Eng. 25 (7) (2010) 530–545.

[53] N. Srivastava, Improving neural networks with dropout, University of Toronto, 2013 Ph.D. thesis.

[54] D.C. Liu, J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. 45 (1–3) (1989) 503–528.

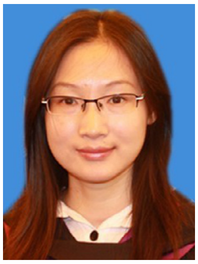
[55] W. Hu, L. Yan, K. Liu, H. Wang, A short-term traffic flow forecasting method based on the hybrid PSO-SVR, Neural Process. Lett. 43 (2015) 1–18.

[56] J.Z. Zhu, J.X. Cao, Y. Zhu, 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 (2014) 139–154.

[57] T. Kkdeniz, Least squares boosting algorithm on short term load forecasting, in: Proceedings of the 8th Ege Energy Symposium And Exhibition, 2016 116–116.

Teng Zhou received the B.E. degree of information and computing science from South China Normal University, Guangzhou, China, in 2010 and M.E. degree of computer technology from Sun Yat-Sen University, Guangzhou, China, in 2012. He is currently working toward the Ph.D. degree in computer application technology at School of Computer Science and Technolgy, South China University of Technology, Guangzhou, China. His research interests include intelligent transportation systems and machine learning and its applications.

Guoqiang Han is a professor at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is the head of the School of Computer Science and Engineering in SCUT. He received his B.Sc. degree from the Zhejiang University in 1982, and the Master and Ph.D. degree from the Sun Yat-Sen University in 1985 and 1988, respectively. His research interests include multimedia, computational intelligence, machine learning and computer graphics. He has published over 100 research papers.

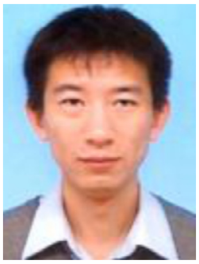


Xuemiao Xu received her B.S., M.S. degrees from South China University of Technology and Ph.D. degree from the Chinese University of Hong Kong in 2002, 2005 and 2009, respectively. She is currently a professor in School of Computer Science and Engineering, South China University of Technology. Her research interests include computer graphics, non-photorealistic rendering, and intelligent transportation based on image analysis.

Zhizhe Lin received her M.S. degrees from Guangzhou Medical University. Her research interests include computer image processing, speech recognition algorithm.

Chu Han graduated from the South China Agricultural University in 2011 with a B.Sc. degree in Computer Science. He received the M.Phil. degrees in computer science from the South China University of Technology in 2014, under the supervision of Prof. Xu Xuemiao. Now he is pursuing Ph.D. in the Department of Computer Science and Engineering of the Chinese University of Hong Kong, under the supervision of Prof. Wong Tien-Tsin. His current research interests include computer graphics, image processing, pattern recognition and computer vision. (chan@cse.cuhk.edu.hk)

Yuchang Huang is currently an undergraduate in the Department of Computer Science and Technology, South China Agricultural University. Her research interests include computer image processing, speech recognition algorithm.



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