2018 IEEE 7th Data Driven Control and Learning Systems Conference
May 25-27, 2018, Enshi, Hubei Province, China

Short-Term Traffic Flow Prediction Based on XGBoost

Xuchen Dong[1], Ting Lei[1], Shangtai Jin[1],Zhongsheng Hou[1]

1. Advanced Control Systems Lab, Beijing Jiaotong University, Beijing, China, 100044

E-mail: 17120217@bjtu.edu.cn, 14111049@bjtu.edu.cn, shtjin@bjtu.edu.cn, zhshhou@bjtu.edu.cn

Abstract: Fast and accurate short-term traffic flow prediction is an important precondition for traffic analysis and control. Due to the fact that the short-term traffic flow has nonlinear characteristic and changes randomly, concurrent computation is difficult for traditional machine learning algorithms. In this paper, a traffic flow prediction model combining wavelets decomposition and reconstruction with the extreme gradient boosting (XGBoost) algorithm is proposed to predict the short-term traffic flow. First, in the training part, wavelet de-noising algorithm is utilized to obtain the high and low frequency information of target traffic flow. Secondly, the high frequency information of traffic flow is processed by threshold method. After that, the high and low frequency information is reconstituted as the training label. Finally, the de-noised target flow is sent to the XGBoost algorithm for training to predict traffic flow. In this way, the trend of the traffic flow in each sample period is retained, and the influence of the short-term high frequency noise is reduced. The proposed traffic flow prediction method is tested base on the traffic flow detector data collected in Beijing, and the proposed method is compared with support vector machine (SVM) algorithm. The result shows that the prediction accuracy of the proposed algorithm is much higher than SVM, which is of great importance in the field of traffic flow prediction.

Key Words: Short-term traffic prediction, wavelet de−noising, XGBoost

1. Introduction

With the development of society, the number of vehicles is increasing rapidly, and the urban traffic congestion problem is becoming more and more serious. As the basis of the intelligent transportation system (ITS), accurate and rapid prediction of traffic flow has become the vital key to solve the congestion problem. In the past decades, short-term traffic prediction has been a hot issue in ITS, and a lot of results were obtained.

So far, several traffic flow prediction methods have been proposed, such as time series analysis [1], grey prediction theory [2], chaos theory [3], Kalman filtering [4], and so on. These methods are difficult to handle the uncertainty and non-linearity characteristic of traffic flow. Another kind of methods is based on artificial intelligence technology, such as support vector machine (SVM) method [5], KNN (k-Nearest Neighbor) method [6], BP neural network method [7], Bayes network [8]. These methods ensure that higher-dimensional non-linear function can be fitted. However, over-fitting occurs when too many parameters are included, and it leads to a low training efficiency if the training samples are too many. It is difficult for existing single method to reflect the periodicity, non-linearity and randomness of the traffic flow. Therefore, many combined models have been put forward in recent years. For example, the SVM method is combined with the particle swarm optimization model, and an intelligent algorithm is utilized to obtain a global optimal solution when determining SVM parameters [9]. In this way, higher prediction accuracy could be obtained, however, it increases not only the complexity of the algorithm, but also the risk of overfitting. CNN neural network is combined with the SVM method in [10]. It replaces the softmax method in CNN's output layer with SVR method and obtains the useful information of traffic flow data.

There are many kinds of machine learning algorithms applied in the transportation field, however, with the increasing amount of traffic data, these methods always need huge computing resources and long computing time, and may lead over-fitting. In this paper, a short-term traffic flow prediction method is proposed based on the Extreme Gradient Boosting (XGBoost) [11], the wavelet decomposition and the threshold method to address the abovementioned problem. XGBoost is an improvement of Boosted Decision Tree algorithm. In terms of precision, XGBoost is proposed to approximate the loss function by using second order Tailor expansion, which uses the first and second derivative information of the loss function. In addition, In order to prevent over-fitting, regularization parameters are added to the original target function of the Boosted Decision Tree algorithm. In terms of calculation, XGBoost improves the efficiency of the algorithm by sorting the traffic data according to the feature values before starting the calculation, and realizing parallel computing on feature enumerations. The wavelet decomposition is utilized to extract periodic characteristics from traffic flow data, and threshold method is used to eliminate the high frequency noise. After that, this information is put into the XGBoost to predict traffic flow. While ensuring the accuracy and robustness of the method, the operation efficiency of the XGBoost method is greatly enhanced.

This paper is organized as follows: Section 2 describes the short-term traffic flow. In Section 3, the short-term traffic flow prediction method is presented based on wavelet decomposition algorithm, threshold algorithm and XGBoost algorithm. In Section 4, the presented method is verified by simulations. The conclusions are given in Section 5.

2. Problem Formulation

In this paper, the short-term traffic flow of a specified lane is predicted by collecting the traffic information from

*This work is supported by National Nature Science Foundation under Grant (61573054, 61433002, 61573129).

978-1-5386-2618-4/18/$31.00 ©2018 IEEE DDCLS'18

traffic detectors, which could be described as follows

$$\begin{aligned}\hat{V}_{cur}(t+1) = F[&t, \varepsilon_c, \varepsilon_l, V_{cur}(t), v_{cur}(t), o_{cur}(t), \\ &c_{cur}(t), V_{pre}(t), v_{pre}(t), o_{pre}(t), c_{pre}(t)],\end{aligned} \quad (1)$$

The meaning of each parameter in (1) is shown in Table 1, and the location of detectors is depicted in Fig. 1.

Table 1: Parameters in (1)

$\hat{V}_{cur}(t+1)$	predicted traffic flow
t	time
ε_c	Cross-section number,
ε_l	the lane number of current section,
$V_{cur}(t)\ (veh/h)$	traffic flow of current lane at time t,
$V_{pre}(t)\ (veh)$	traffic flow of previous lane at time t,
$v_{cur}(t)\ (km/h)$	the average speed of vehicles in current lane at time t,
$v_{pre}(t)\ (km/h)$	average speed of vehicles in previous lane at time t,
$o_{cur}(t)\ (\%)$	occupancy of current lane at time t,
$o_{pre}(t)\ (\%)$	occupancy of previous lane at time t,
$c_{cur}(t)\ (veh)$	cart number in current lane at time t,
$c_{pre}(t)\ (veh)$	cart number of previous lane at time t,
$V(t)$	Training label.

There are two steps to apply the abovementioned short-term traffic flow prediction model: training and prediction, respectively.

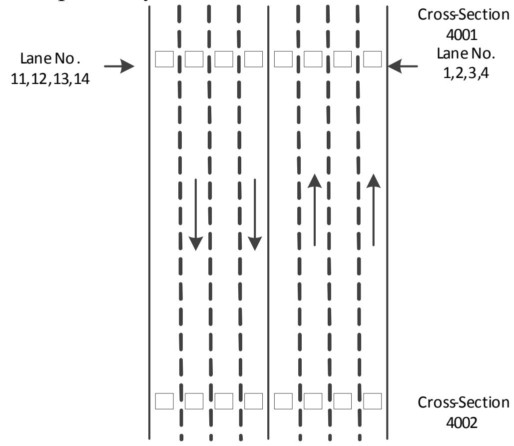

Fig. 1: location of detectors

In the training step, the monitoring data of 30 monitoring sections of the Fourth Ring Road in Beijing, is selected as training samples. Each monitoring point detects the number of vehicles and carts, and the average speed and occupancy for each lane. The monitoring data from June 24 to 29, 2008, is chosen as the training set, while the data in June 30 is the testing set. There are 862859 training samples and 153095 test samples to train and test the short-term traffic flow prediction model. Some samples are shown in Table 2.

Table 2: Part of samples

Cross-Section Number	Lane Number	Flow (veh/2min)	Speed (km/h)	Cart (veh/2min)	Occupancy (%)	Hours	Minutes	Days	Object Flow
4002	14	21	63	4	3	0	0	24	24
4002	13	9	76	0	1	0	0	24	29
4002	12	12	77	1	1	0	0	24	24
4002	11	6	94	1	0	0	0	24	20

Lane Number (1, 2, 3, 4 means lanes from east to west or from south to north, and 11, 12, 13, 14 means lanes from west to east or from north to south, lane 1 and 11 are near the center). Traffic flow data from No.4002 detector in lane 1 on June 24 are shown in Fig. 2.

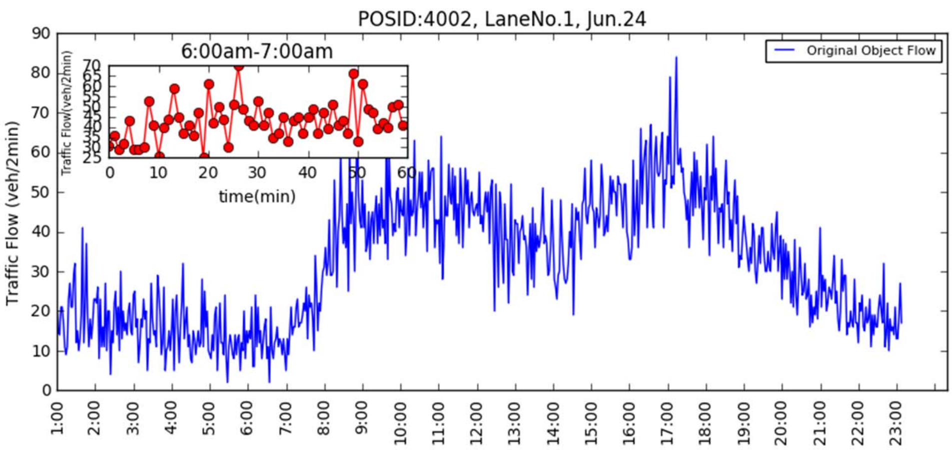

Fig. 2: Part of samples

In view of the above characteristics of the monitoring data, this work deals with the training label $V(t)$ by wavelet decomposition and reconstruction. Firstly, the high frequency noise after decomposition is handled by threshold. After that, it is reconstructed with low frequency traffic flow information. In this way, the long-term regularity of traffic flow can be retained, and the high frequency noise is eliminated, which makes the prediction model more robust. Secondly, the training label $V(t)$ is put into XGBoost model as the training label. The best model parameters are obtained when the target precision is achieved. The flow chart of model training and prediction is shown in Fig. 3.

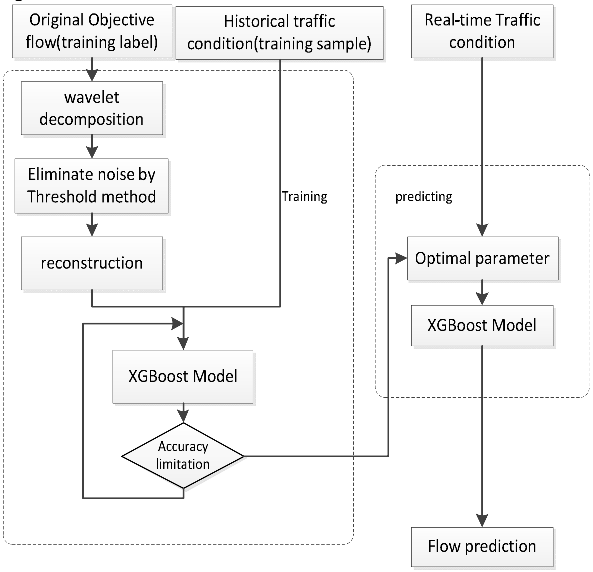

Fig. 3: Structure of the training and prediction model Algorithm

3. Methodology

3.1. Discrete Wavelet De-noising Algorithm

In order to improve the model robustness and handle the randomness of the short-term traffic flow, the wavelet decomposition and the reconstruction are utilized to de-noise before traffic flow prediction.

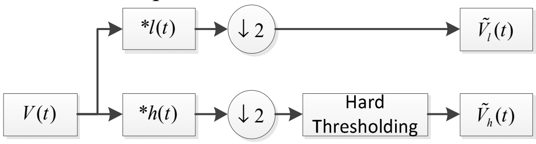

Fig. 4: Wavelet decomposition

The wavelet decomposition process is shown in Fig. 4, where * is the convolution formula, $\downarrow 2$ is the decimation factor of 2. $h(t)$ and $l(t)$ represent high-pass and low-pass filters, which is defined as follows

$$h(t)=\begin{cases}\dfrac{1}{\sqrt{2}} & \text{if } t\text{=0,1}\\ 0 & otherwise\end{cases} \tag{2}$$

$$l(t)=\begin{cases}\dfrac{1}{\sqrt{2}} & \text{if } t\text{=0}\\ -\dfrac{1}{\sqrt{2}} & \text{if } t\text{=1}\\ 0 & otherwise\end{cases} \tag{3}$$

$\tilde{V}_h(t)$ and $\tilde{V}_l(t)$ are the high and low frequency information of traffic flow, which are calculated by the following equations

$$\tilde{V}_h(t)=(V*h)(t)=\sum_{k=0}^{N-1}V(k)h(2t-k) \tag{3}$$

$$\tilde{V}_l(t)=(V*l)(t)=\sum_{k=0}^{N-1}V(k)l(2t-k) \tag{4}$$

The high and low frequency information is further down-sampled by a factor of 2. Then the high frequency information is handled by threshold.

$$\tilde{V}_h(t)=\begin{cases}\tilde{V}_h(t) & \text{if } \tilde{V}_h(t)<1\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

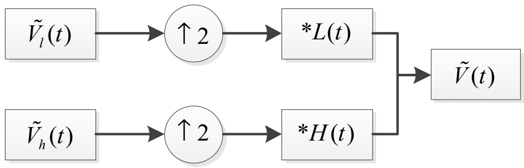

Fig. 5: Reconstruction

After the wavelet decomposition process, the high and low frequency information is reconstructed. Reconstruction is the inverse process of decomposition. After up-sampling process of the high and low frequency information, the new training label $\tilde{V}(t)$ is obtained by convolving the inverse transformation of high and low pass filter with the coefficients. The process of wavelet reconstruction is described in Fig. 5, where $\uparrow 2$ means up-sampling by a factor of 2, and the new training label $\tilde{V}(t)$ is reconstructed as follow.

$$\begin{aligned}\tilde{V}(t)&=(\tilde{V}_l*L)(t)+(\tilde{V}_h*H)(t)\\ &=\sum_{k=0}^{N-1}\tilde{V}_l(k)L(t-k)+\sum_{k=0}^{N-1}\tilde{V}_h(k)H(t-k)\end{aligned} \tag{6}$$

where

$$H(t)=\begin{cases}\dfrac{\sqrt{2}}{2} & \text{if t=0,1}\\ 0 & otherwise\end{cases} \tag{7}$$

$$L(t)=\begin{cases}\dfrac{\sqrt{2}}{2} & \text{if t=0}\\ -\dfrac{\sqrt{2}}{2} & \text{if t=1}\\ 0 & otherwise\end{cases} \tag{8}$$

3.2. Boosted Tree and XGBoost

The prediction algorithm proposed in this paper is based on XGBoost, which is improved version of Boosted Tree.

3.2.1. Boosted Tree

Boosting Tree is a boosting learning algorithm using linear combination of multiple weak "learners" to improve the learning accuracy of the algorithm. Regression tree is also a basal learner to solve regression problem.

Assume that the training samples $x(t)$ includes n observations, where each observation is a m-dimensional vector, training label $V(t)$ also includes n observations, and $D=\{(x(t),V(t))(|D|=n,x(t)\in R^m,V(t)\in \mathrm{R})\}$. In this paper, $x(t)$ as the m-dimensional vector, is the traffic data sampled at time t, which contains the traffic flow information mentioned in Section 2. First of all, in order to find the best split feature j and best split point s_j, all the features are traversed to scan the possible split point s_j for each feature j by solving the following problem:

$$\min_{j,s_j}[\min_{c1}\sum_{x(t)\in R_1(j,s_j)}^{n}(V(t)-c_1)^2+\min_{c1}\sum_{x(t)\in R_2(j,s_j)}^{n}(V(t)-c_2)^2] \tag{9}$$

$$R_1(j,s_j)=\{x(t)\,|\,x^{(j)}(t)\le s\},\ R_2(j,s_j)=\{x(t)\,|\,x^{(j)}(t)>s\} \tag{10}$$

$$c_k=\frac{1}{N_m}\sum_{x(t)\in R_1(j,s_j)}V(t),\ x(t)\in R^m,\ k=1,\ 2 \tag{11}$$

where $x^j(t)$ is the feature j of $x(t)$, $R(j,s_j)$ represents the division area of the j-th feature, and c_k represents the sample average of the k-th area. After splitting, (9), (10), (11) are iterated until the accuracy requirement is satisfied.

Therefore, the regression tree model can be written below:

$$f=w_{q(x(t))} \tag{12}$$

where q is the structure of each tree that maps a sample to the corresponding leaf index, and w is the weight of the leaf.

For a given regression tree model, the objective function of Boosting Tree model can be rewritten as follow,

$$obj(q,w)=\text{argmin}\sum_{t=1}^{n}l\left(V(t),\hat{V}_{cur}^{(p)}(t+1)\right) \tag{13}$$

where $\hat{V}^{(p)}(t+1)$ is the prediction traffic flow obtained by the additional model at step p, and l represents the loss function. The additional model at step p is described below:

$$\hat{V}_{cur}(t+1)^{(p)}=\hat{V}(t+1)^{(p-1)}+f^{(p)}(x(t)) \tag{14}$$

Because the basic classifiers of the additional model are the regression tree, the objective function is presented below:

$$obj(q,w)=\text{argmin}\sum_{t=1}^{n}l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)+f^{(p)}(x(t))\right) \tag{15}$$

Solving the regression problem, the loss function in the objective function is always defined as squared error loss function. The objective function can be rewritten as follow:

$$obj(\text{q},\text{w})=\text{argmin}\sum_{t=1}^{n}\left(V(t)-\hat{V}_{cur}^{(p-1)}(t+1)-w_{q(x(t))}^{(p)}\right)^2 \tag{16}$$

3.2.2. XGBoost

In order to constrain the number of leaf nodes and weights, regularization is added to the objective function of the Boosting Tree model in XGBoost algorithm. The regularized objective function at the p-th iteration is presented below:

$$obj(q,w)=\text{argmin}\left(\sum_{t=1}^{n}l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)+f^{(p)}(x(t))\right)+\Omega\left(f^{(p)}\right)\right) \tag{17}$$

where $\Omega\left(f^{(p)}\right)=\gamma T+\frac{1}{2}\lambda\sum_{g=1}^{T}w_g^2$, T is the number of leaf node, γ and λ are the regularization parameters. For general situation, the loss function is deduced by second order Tailor expansion below:

$$\begin{aligned}obj(q,w)&\approx\text{argmin}(\sum_{t=1}^{n}[l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)\right)\\&+g_tf^{(p)}(x(t))+\frac{1}{2}h_t(f^{(p)}(x(t)))^2]+\Omega(f^{(p)}))\\&=\text{argmin}(\sum_{t=1}^{n}[l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)\right)+g_tw_{q(x(t))}\\&+\frac{1}{2}h_tw_{q(x(t))}^2]+\gamma T+\frac{1}{2}\lambda\sum_{g=1}^{T}w_g^2)\end{aligned} \tag{18}$$

where $g_t=\partial_{\hat{V}^{(p-1)}(t+1)}l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)\right)$, $h_t=\partial^2_{\hat{V}^{(p-1)}(t+1)}l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)\right)$, and the loss function $l\left(V(t),\hat{V}_{cur}^{(p-1)}(t+1)\right)$ of previous iteration is constant. Therefore, we can remove this constant when minimizing the objective function. Define $I_j=\{t\,|\,q(x(t)=j)\}$ as the representation of the j-th leaf node, gives

$$obj(q,w)=\arg\min(\sum_{j=1}^{T}[(\sum_{t\in I_j}g_t)w_j+\frac{1}{2}(\lambda+\sum_{t\in I_j}h_t)w_j^2]+\gamma T) \tag{19}$$

and (20) can be rewrote by denoting $G_i=\sum_{t\in I_j}g_t$, $H_t=\sum_{t\in I_j}h_t$. And for a fixed structure $q(x(t))$, the optimal weight w_j^* of leaf j is computed below:

$$w_j^*=-\frac{G_j}{H_j+\lambda} \tag{20}$$

and the maximum gain of the corresponding objective function is obtained as follow.

$$\min Obj=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T \tag{21}$$

The best structure of regression tree which minimizes the objective function by enumerating different kinds of structure of tree can be obtained from (20)–(22). Due to it is impossible to enumerate all the structures; the XGBoost method tries to add branches for every existing leaf nodes. The loss reduction after the split is given below.

$$Gain=\frac{1}{2}[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}]-\gamma \tag{22}$$

where $\frac{G_L^2}{H_L+\lambda}$ is the gain of the optimal w, which is calculated by left node after split. $\frac{G_R^2}{H_R+\lambda}$ is the gain from right node. $\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}$ is the gain if the split is not added. Finally, the prediction flow $\hat{V}(t+1)$ is calculated after p th iteration, see (15).

4. Simulation and Analysis

In this section, features are selected firstly, because prediction result will be greatly influenced by features, which fed into the prediction method. Furthermore, processing of training label will be introduced which processed by wavelet decomposition and reconstruction. At last, the prediction result will be compared with SVM method and XGBoost method without wavelet processing.

4.1. Feature Selection

In this paper, the basic traffic flow information obtained by detectors is selected as the data feature, which includes the number of vehicles and carts, time occupancy, and average speed. The trend of traffic flow from different sections at different times in every day is periodical. Therefore, time and the number of lanes and sections are added into the traffic data feature. Besides, traffic flow information from the previous section is also added, because the information from previous section could reflect the trend in current section. The comparison flow between section No.4001 and No.4002 at 4:00pm on 24th is shown in Fig. 6.

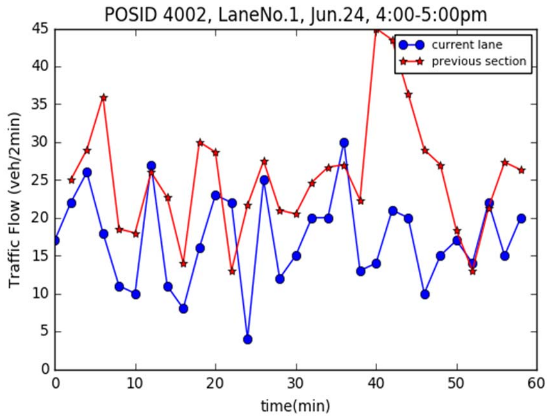

Fig. 6: Comparison between current and previous traffic flow

As can be seen from Fig. 6, the trend of traffic flow in current section is the offset value of the previous section, that is, the information from previous section could reflect the trend from current section next time.

4.2. Preprocessing

Since regression tree is the basic learner of XGBoost, there is no need to normalize samples, which means that features from different units would not affect the prediction result.

The labels in training samples and the high frequency noise are processed by wavelet decomposition threshold method. Respectively, the result after reconstruction is shown in Fig. 7.

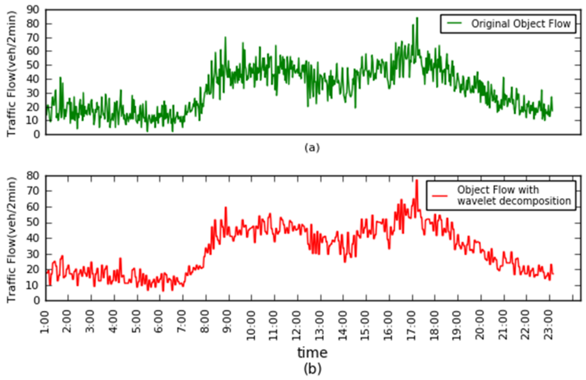

Fig. 7 The result after wavelet decomposition and reconstruction

4.3. Model training and result analysis

The parameters of XGBoost are searched by grid, and the final parameters are determined as follows: The number of basic learners is 100, the max depth for each regression tree is 12, the learning rate is 0.3, and the proportion of random sampling for each tree is 0.5.

Fig. 8 and Fig. 9 show the result for part of comparison of the observation traffic flow data and prediction value of No. 4024 detector in the test sample.

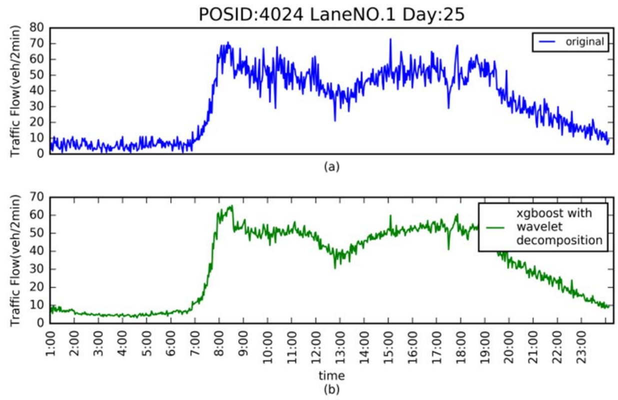

Fig. 8: observation value and prediction value

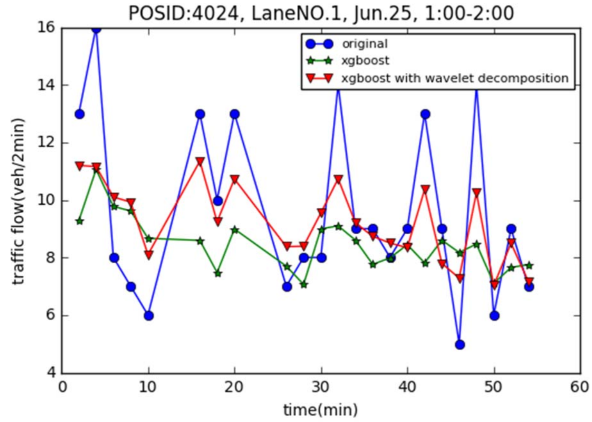

Fig. 9 observation value and prediction value from 1:00 to 2:00

In Fig. 8 and Fig. 9, it can be seen that the prediction method has a more accurate result of long-term traffic flow, and short-term prediction results are smoother.

In this paper, the Root Mean Square Deviation (RMSE) and Mean Absolute Percentage Error (MAPE) are used to analyze and evaluate the model performance, which are shown as follows:

$$e_{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \tag{23}$$

$$e_{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \tag{24}$$

Where training label y_i and prediction value $\hat{y}_i$ represent the observation and prediction value of samples, respectively, and n is the size of samples.

The results of XGBoost and SVM model are shown in Table 3. It can be seen that both of RMSE and MAPE are reduced after the wavelet decomposition and reconstruction, and the operating efficiency of XGBoost is much better than SVM model.

Table 3: Results for different methods

	SVM	XGBoost	XGBoost with wavelet decomposition
RMSE	7.306087	6.180097	6.117778993
MAPE	0.305323	0.22145	0.212503699

5. Conclusions

This paper presents a short-term traffic prediction method combining wavelet decomposition and reconstruction with XGBoost. Wavelet decomposition and reconstruction not only retain useful information, but also reduce the influence of randomness of traffic information on the prediction model. Furthermore, XGBoost is used to predict the traffic flow, which improves the precision and efficiency of the prediction model. Finally, the simulation results show that training accuracy is improved.

References

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

[2] J. Yang, X. Xiao, S. Mao, et al, Grey coupled prediction model for traffic flow with panel data characteristics, *Entropy,* 18(12):454, 2016.

[3] C. Dong, Z. Liu, Z. Qiu, Prediction of traffic flow in real-time based on chaos theory, *Information & Control*, 33(5): 518–522, 2004.

[4] W. Huang, W. Jia, J. Guo, et al, Real-time prediction of seasonal heteroscedasticity in vehicular traffic flow series , *IEEE Trans. on Intelligent Transportation Systems*, Online, 2017.

[5] Q. Meng, Small-time scale network traffic prediction based on a local support vector machine regression model, *Chinese Physics B*, 18(6):2194–2199, 2009.

[6] M. May, D. Hecker, C. Körner, et al, A vector-geometry based spatial knn-algorithm for traffic frequency predictions, in *Data Mining Workshops. ICDMW'08. IEEE International Conference on. IEEE*, 2008: 442–447.

[7] X. Guo, F. Deng, Short-term prediction of intelligent traffic flow based on BP neural network and ARIMA model, in *E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on. IEEE*, 2010: 1–4.

[8] S. Sun, C. Zhang, G. Yu, A bayesian network approach to traffic flow forecasting, *IEEE Trans. on Intelligent Transportation Systems*, 7(1):124–132, 2006.

[9] M. Duo, Y. Qi, G. Lina and E. Xu, A short-term traffic flow prediction model based on EMD and GPSO-SVM, in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017: 2554-2558.

[10] W. Luo, B. Dong, Z. Wang, Short-term traffic flow prediction based on CNN-SVR hybrid deep learning model, *Journal of Transportation Systems Engineering and Information Technology*, 17(5): 68–74, 2017.

[11] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM*, 2016: 785-794.