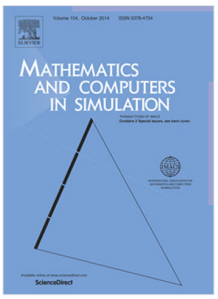
Journal Pre-proof



An interpretable model for short term traffic flow prediction

Wei Wang, Hanyu Zhang, Tong Li, Jianhua Guo, Wei Huang,
Yun Wei, Jinde Cao

PII: S0378-4754(19)30372-6
DOI: https://doi.org/10.1016/j.matcom.2019.12.013
Reference: MATCOM 4917

To appear in: *Mathematics and Computers in Simulation*

Received date : 6 July 2019
Revised date : 17 October 2019
Accepted date : 18 December 2019

Please cite this article as: W. Wang, H. Zhang, T. Li et al., An interpretable model for short term traffic flow prediction, *Mathematics and Computers in Simulation* (2019), doi: https://doi.org/10.1016/j.matcom.2019.12.013.

An interpretable model for short term traffic flow prediction

Wei Wang[a], Hanyu Zhang[a], Tong Li[b], Jianhua Guo[c], Wei Huang[c], Yun Wei[d], Jinde Cao[c,*]

[a]*Software School, Yunnan University, Kunming 650091, China*
[b]*School of Big Data, Yunnan Agricultural University, Kunming 650201, China*
[c]*School of Mathematics, Southeast University, Nanjing 210018, China*
[d]*National Engineering Laboratory for Green and Safe Construction Technology in Urban Rail Transit, Beijing 100037, China*

Abstract

Predicting short term traffic flow to improve traffic control is a research problem attracting increased attention over the past 30 years. With increasing number of traffic data acquisition equipments coming into usage, it provides an opportunity to use deep neural network (DNN) to predict short-term traffic flow. Behind its considerable success, the DNN is weighed down by some problems, and here we focus on: 1. how to justify the number of input nodes employed by DNN; 2. how to explain the causality between the historical spatiotemporal information and the future traffic condition. In this paper, we propose a deep polynomial neural network combined with a seasonal autoregressive integrated moving average model. The new model has superior predicting accuracy as well as enhanced clarity on the spatiotemporal relationship in its deep architecture. Experimental results indicate that the proposed model has better explanation power and higher accuracy compared with the LSTM based model.

Keywords: Deep architecture, GMDH, short term traffic flow prediction

*Corresponding author
Email addresses: wangwei@ynu.edu.cn (Wei Wang), 745269194@qq.com (Hanyu Zhang), tli@ynu.edu.cn (Tong Li), seugjh@163.com (Jianhua Guo), seuhwei@126.com (Wei Huang), luckyboy0309@163.com (Yun Wei), jdcao@seu.edu.cn (Jinde Cao)

1. Introduction

Predicting the short term traffic flow is a problem attracting increased attention for more than 30 years [1]. Success in prediction leads to improved traffic operation efficiency which benefits individual travellers, business, government agencies and more. With more and more equipments coming to use [2], the variety and volume of traffic data that could be adopted for short term traffic flow prediction grew rapidly. Traffic management bureaus, loop detectors, sensors and controllers placed on the roadways, video cameras, GPS-enabled vehicles, mobile devices of individuals all contribute to the richness of the traffic data. With the easy access of the traffic data, researchers have developed many data driven prediction methods built on time series models [3, 4, 5], Markov chain model [6], Kalman filter theory [7, 8], local regression models [9], spectral analysis [10], non-parametric methods[11, 12], Bayesian networks [13, 14], and neural networks [15, 16, 17]. It has become the most dynamic research area [18, 19]. At the same time, their performances still has much room to improve.

Recently, deep stacked autoencoder [2], long short term memory neural network (LSTM)[20] and other deep learning architectures are adopted into this research area and produced many encouraging results. However, the advance of the development is slowed down by the issues of interpreting the relationship between the prediction results and the spatiotemporal information.

In practice, many countries actively develop their intelligent transportation system (ITS). During the development process, it is gradually clear that accurate prediction is only a part of ITS's mission. An explicit representation between the prediction results and the spatiotemporal information is also helpful in decision making to the traffic managers of ITS. However, unlike many traditional models (such as time series models, Kalman filter) which assign the optimal weights to the human interpretable features to clarify the causality between inputs and prediction results, the behavior of deep learning models is much less easily interpreted [21]. Deep learning models mainly operate on inputs through multi-layer neural architectures. Each layer of which is characterized as an array

of hidden neuron units. It is unclear how deep learning models assign weights to hidden neuron units, and why some parts of hidden units are combined and other parts of hidden units are dropped out [22].

So, there is an urgent demand for a deep learning architecture that (a) can accurately predict the future traffic flow; (b) justify the number of input nodes employed; and (c) explain the causality between the historical spatiotemporal information and the future traffic conditions [18].

To meet these challenges, we propose a deep polynomial neural network called Group Method of Data Handling (GMDH) plus the Seasonal Autoregressive Integrated Moving Average (SARIMA) model. The main contributions of the proposed model are listed below.

1. the short-term traffic flow is decomposed into two parts: trend and residual. The residuals are used as the input to GMDH that improved the precision of prediction and make the GMDH converge quickly.

2. the number of input nodes is determined by the SARIMA model.

3. with the help of the self-organizing feature of GMDH, the casualty between the historical spatiotemporal information and the future traffic conditions is presented as an explicit form, a polynomial. In other words, the future traffic flow is interpreted as a polynomial function of the historical temporal and spatial traffic information.

The organization of this paper is as follows: Section two gives a literature survey related to our paper. Section three presents the technical details of our approach. Traffic flow data used in the empirical study and the empirical results are presented in Sections four and five. The discussions and conclusions are in Section six.

2. Literature review

In this section, we briefly review the literature of the short term traffic flow prediction. Suppose we have traffic flow data X_i^t at time t from the ith sensor in some traffic network with $i = 1, 2, \ldots, M$ and $t = 1, 2, \ldots T$. We wish to

predict the traffic flow of the i^{th} sensor at the time interval $(T + \Delta)$ over the time increment or prediction horizon Δ.

The first traffic flow predictor was proposed in [23] in 1979. Since then, researchers in various disciplines such as computer science, mathematics, and transportation economy proposed many predictors. These predictors are roughly either data-driven or model-driven [24]. Model-driven predictors use graphs to represent the traffic network topology. The nodes and edges in the graph represent junctions/intersections and roads, respectively. These methods predict future traffic flows by simulating the vehicle movements on the graph. With both the variety and volume of the traffic data growing rapidly, the data-driven approaches are increasingly popular. The data-driven approaches can be further divided into two main subcategories: parametric and nonparametric [20]. Autoregressive moving average (ARMA) time series algorithm is the representative of the parametric method, and it has received sustained attention [25, 3]. Under the ARMA model, the means and the variances of the traffic flow are assumed to be stationary or close to constants over time. However, the mean and variance of the real traffic flow usually change over time, and it does not consist with the ARMA assumption. To overcome this deficiency, transformations are introduced to make non-stationary traffic flows into stationary ones.

Along this line of thinking, ARMA models are extended in many directions. One of them is the autoregressive integrated moving average (ARIMA). In this approach, an integration operator (I) is introduced to handle the flow difference between adjacent time intervals; a seasonal autoregressive integrated moving average (SARIMA) is used to remove the seasonal effect by seasonal difference. See [3, 4, 26] for examples of such algorithms.

The real world traffic flow is heterogeneous: its variance varies over time [27]. To handle heteroscedasticity, the traffic flow variance is often modeled as a simple quadratic function of traffic flows in the precedent time intervals [28]. The generalized autoregressive conditional heteroscedasticity (GARCH) model is hence proposed [29, 30, 31]. The GARCH model is usually combined with other models (such as ARIMA, SARIMA) to predict the uncertainty of the short

term traffic flow leading to predictors with improved performance [32, 33, 10]. As noted in [34, 35, 10], traffic flow has natural spatiotemporal constraints, and is sharply nonlinear, non-stationary and volatile. To overcome the difficulties caused by these features, K-nearest Neighbor (KNN) [36, 37], Hidden Markov Model[6], Support Vector Regression (SVR) [38], particle filter[39], Gaussian Process [40], and other nonparametric algorithms are developed. Most of these algorithms merely search for shallow traffic features with limited information and fail to capture hidden or implicit traffic correlations [20].

The deployment of the traffic data acquisition equipments in the traffic network in recent years leads to traffic data explosion. The traffic data are collected from many sources such as Traffic management bureaus, loop detectors, sensors, controllers [41]. The traffic data are complex, containing implicit traffic information such as spatial, temporal, spatiotemporal correlations, and displaying multi-state features. Deep neural networks (DNN) are renowned at extracting complex features from a massive amount of data without resorting to prior knowledge via multiple-layer architectures. This characteristic ensures the deep learning algorithms excellent performance in traffic flow prediction.

The stacked autoencoder (SAE) based predictor is first employed by [2] for traffic flow prediction. The deep autoencoder takes generic traffic flow features as input but ignores the spatiotemporal correlation. Meanwhile, other deep architectures use spatiotemporal correlation instead of generic traffic flow as input. [42] employed the convolutional neural network (CNN) and obtained a respectable predictor. In this model, the spatiotemporal correlations are transformed into a time-space matrix. [43] employed the spatiotemporal recurrent convolutional network (SRCN). The SRCN combines the advantages of deep convolutional neural networks (DCNN) and long short term memory (LSTM) neural network to capture the spatiotemporal dependencies and temporal dynamics of network-wide traffic. Besides the generic traffic flow or the spatiotemporal correlation, the residual of traffic flow is another kind input to DNN. [44] proposed a deep architecture, DeepTrend, for short term traffic flow prediction. DeepTrend has an extraction layer and a prediction layer. Experiments

show that DeepTrend markedly improves the prediction performance over the traditional prediction approaches.

3. The proposed approach

We propose the GMDH combined with SARIMA and GARCH to enhance casualty explanation and meet spatiotemporal modeling requirements of the short term traffic prediction. Given a generic traffic flow series X, it can be model as $trend + residue$. The trend is a relatively stable temporal pattern in the traffic flow. However, the residuals are the random fluctuations around the trend and are hard to predict. Therefore, the key of the short term traffic flow prediction becomes the prediction of residual sequence. In this paper, the residual series from different local sensors are fed into the GMDH model separately. The trained GMDH model is then used to predict the future traffic flow residue series. The GARCH and SARIMA models are employed to adjust the usually non-stationary residual series. The periods to lag of each residual series is determined by the SARIMA model, and the number of input nodes needed for the GMDH is equal to the sum of periods to lag of each residual series.

The GMDH is the first deep learning architecture proposed by [45]. It is most renowned for its self-organizing feature of arriving at the optimal structure and for its unbiased external criteria for neural selection. These features of GMDH make the predicted future traffic residual flow as a polynomial function of the historical traffic residual conditions presented in Eq. (1) which greatly enhances the explanatory power:

$$R_i^{t+1} = f_\omega \left(R_1, R_2, \ldots, R_n\right) \tag{1}$$

where ω is the parameters in GMDH, f is the polynomial generated by GMDH, R_i is the residual series at the i^{th} local sensors, and R_i^{t+1} is the residue at the i^{th} sensor in the $t+1$ time interval.

We will provide more details in the next a few subsections.

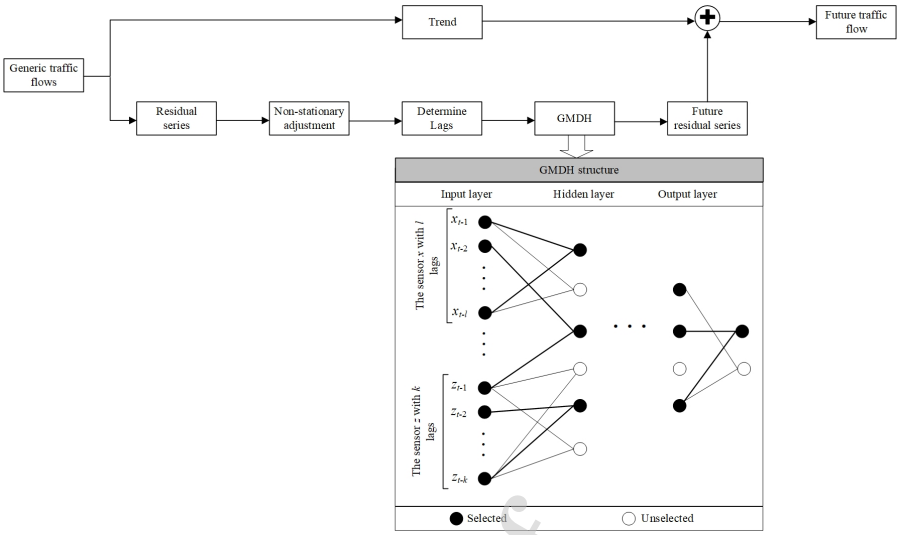


Figure 1: The flow chart of the interpretable short term traffic flow prediction model

3.1. Detrending

The traffic flow series can often be well decomposed into the intra-day trend and the residue series [46]. We first use a detrend operation to remove the trend from the traffic flow series. A model is then learned only on residue series similar to [47, 48, 44]. We observe that detrending also reduces the non-stationarity in the short term traffic flow series.

Simple average is found to be a very effective detrend operation in [47] and it significantly improve the performance of short term traffic flow prediction. Following their lead, we also use the simple average based intra-day detrend operation.

Let $y_i^{w,t}$ be the w^{th} observed traffic flow at sensor i in the t^{th} time interval. The traffic flow series in W consecutive observations at sensor i are as in Eq. (2).

$$Y_i^1 = \left[y_i^{1,1}, y_i^{1,2}, \ldots, y_1^{1,n}\right], \ldots, Y_i^W = \left[y_i^{W,1}, y_i^{W,2}, \ldots, y_i^{W,n}\right] \tag{2}$$

where n is the number of observations per day. If the observation time interval is 5 minutes, then n=288. The simple average trend at sensor i over past W

days is given by

$$\bar{Y}_i = \frac{1}{W}\sum_{k=1}^{W} Y_i^k. \tag{3}$$

The residue traffic series at the i^{th} sensor in the t^{th} day is given by

$$R_i^t = Y_i^t - \bar{Y}_i. \tag{4}$$

We form the residue series in W consecutive days at the i^{th} sensor as

$$R_i = R_i^1 \cdot R_i^2 \cdot \ldots \cdot R_i^W \tag{5}$$

This above residue series is finally fed into the GMDH instead of the original traffic flow series.

3.2. Determine the number of input nodes

In this section, we describe our method to determine the input nodes of GMDH. In the classical GMDH used for time series prediction, future traffic flow y_t is predicted by $y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-l})$ for some function $f(\cdot)$ and the lagged variables $y_{t-1}, y_{t-2}, \ldots, y_{t-l}$. l is called the order of periods to lag and the number of input nodes for the GMDH is equal to l [10]. To take the spatiotemporal correlation into consideration, the proposed GMDH takes residue traffic flow series from nearby sensors as input. The number of input nodes of the learned GMDH is the sum of periods to lag of these input residue series. This arrangement explicitly reveals their influences in the GMDH neural network.

We use the SARIMA$(p,d,q)(P,D,Q)s$ to model the traffic flow residue series of each nearby sensor. We postulate, for the i^{th} sensor, the residue series R_i^t can be modeled as:

$$\Phi_P(B^s)\phi_p(B)\nabla_s^D\nabla^d R_i^t = \theta_q(B)\Theta_Q(B^s)\varepsilon_i^t \tag{6}$$

where t is the time interval index, B is the backshift operator that $BR_i^t = R_i^{t-1}$, p is the order of the short term autoregressive polynomial, d is the order

of the integrated polynomial and $\nabla^d = (1-B)^d$ the regular differences, q is the order of the moving average polynomial, P is the order of the seasonal autoregressive order, D is the order of the seasonal differencing, Q is the order of the seasonal moving average, s is the number of time steps for a single seasonal period. $\Phi_P(B^s) = (1 - \Phi_1 B^s -, ..., -\Phi_P B^{sP})$ is the seasonal autoregression operator of order P, $\phi_p = (1 - \phi_1 B -, ..., -\Phi_P B^P)$ is the regular autoregression operator of order p, $\nabla_s^D = (1-B^s)^D$ is the seasonal differences, $\Theta_Q(B^s) = (1 - \Theta_1 B^s -, ..., \Theta_Q B^{sQ})$ is the seasonal moving average operator of order Q, $\theta_q(B) = (1 - \theta_1 B -, ..., \theta_Q B^q)$ is the regular moving average operator of order q, ε_i^t is a white noise term.

In the above model, the remaining non-stationary elements after detrending, are adjusted by the integrating and seasonal difference operators of SARIMA model. The autogressive part of SARIMA indicates how many lagged variables are related to the future traffic flow, and the moving average part indicates how the regression error is combined with the errors in the past. Since the moving average part do not contribute to the number of input nodes for GMDH, so we choose $q = 0$ and $Q = 0$. The values of p, d, P, and D in this model are chosen by analyzing the ACF and PACF graphs of the stationary series.

It is conventionally assumed that the error term ε_t^i is the white noise with zero mean and constant variance. However, heteroscedasticity has been proved widely existed in the traffic flow series [27], and the prediction performance can be improved by removing it [10]. We use GARCH model as follows for error terms of ε_i^t:

$$\varepsilon_t^i = \sqrt{h_t e_t}, \tag{7}$$

$$h_t = \alpha_0 + \sum_{i=1}^{v} \alpha_i \left(\varepsilon_t^i\right)^2 + \sum_{i=1}^{u} \beta_i h_{t-1}, \tag{8}$$

$$e_t = IN(0,1) \tag{9}$$

where t is the time interval index, h_t is the conditional variance at t, u is the autoregressive order of GARCH process and $u > 0$, v is the moving average

order of GARCH process and $v > 0$, α_0 is a positive constant coefficient, α_i is the coefficients of the lagged errors, β_i is the coefficients of the lagged conditional variance h_{t-i}, and e_t is the independent normal variable with zero mean and unit variance. The order of u and v can be selected by AIC. However, in practice, many analysts find GARCH(1,1) is suitable for modeling the heteroscedastic time series [4, 49]. This paper also use GARCH(1,1).

Given h local sensors, $\{R_1, R_2, \ldots, R_h\}$ are corresponding residual series from different local sensors. Each residual sequence R_j is modeled by SARIMA and corresponds to one parameter p_j. So, the number of input nodes for GMDH model is equal to $\sum_{j=1}^{h} p_j$.

3.3. GMDH

We construct the GMDH model as follows. We assume that the traffic flow at a location is not related to flows at far away locations, and select the traffic residue flows from the neighboring road sensors as input. The procedure of constructing GMDH predictor involves 5 steps:

Step 1. The residual series from different sensors are separated into training set and testing sets. Then the training data is used to construct the GMDH model and the testing data is used to evaluate the performance of the GMDH model. The number of input nodes for GMDH is equal to $Z = \sum_{j=1}^{h} p_j$

Step 2. Choose the partial description (PD) of the GMDH. Linear, linear covariation and cubic reference functions are most widely used. These functions are defined as below:

$$\text{linear: } y = w_0 + w_1 x_1 + w_2 x_2 \tag{10}$$

$$\text{linear covariation: } y = w_0 + w_1 x_1 + w_2 x_2 \quad + w_3 x_1 x_2 \tag{11}$$

$$\text{cubic: } y = w_0 + w_1 x_1^3 + w_2 x_1^2 + w_3 x_1 + w_4 x_1^2 x_2 + w_5 x_2^3 + w_6 x_2^2 + w_7 x_2 \tag{12}$$

$L = Z(Z-1)/2$ new variables are constructed according to the PD selected.

Step 3. Estimate the corresponding parameters of the PD. The least square is used to determine the parameters of the PD.

Step 4. Determine new input variables for next layer according external criteria. There are numerous external criteria can be used for GMDH [50]. In this paper, we adopt average regularity criterion (ARC).

$$ARC = \frac{1}{N}\sum_{t\in N}(y_t - \hat{y}_t)^2 \tag{13}$$

Where N is the size of testing set, y_t is the traffic residue flow of t^{th} observation, and $\hat{y}_t$ is the prediction of t^{th} observation by the learned GMDH model. The y_t is selected as the new input variable when the ARC of y_t is less than the threshold predefined, otherwise drop y_t.

Step 5. Check the stopping criterion. The lowest value of external criteria using GMDH model obtained at this layer is compared with the smallest value obtained at the previous one. If there is an improvement, one goes back and repeats step 1 to 5, otherwise the iterations will terminate and a realization of the network is completed. Once the final layer is determined, only the one node characterized by the best performance can be selected as the output node. The remaining nodes in that layer will be discarded. Finally, the trained GMDH model is obtained.

4. Empirical study

In this section, four important issues regarding the empirical study are presented. First, the description of the data used in this empirical study is given. Second, the processes of trend and residual series acquisition with different time intervals are presented. Third, the architecture of GMDH-based predictor is presented. Finally, the measurement indexes used to evaluate the performance of the predictor are given. Specifically, we try to answer the following research questions (RQs).

RQ1: To what extent is the prediction accuracy improved by our approach? We set the Long Short Term Memory (LSTM) based predictor as baseline approach and the performance index of such two predictors are compared.

RQ2: To what extent is the interpretability is improved by the self-organizing feature of GMDH? The output of GMDH could be presented as a polynomial of input variables. The interpretability is equal to the causality between input variables and output of the GMDH.

4.1. Data Collection

To guarantee the objectivity of the empirical study, the data used in empirical study contain more than 230,000 vehicle records in the Nanming district of Guiyang, Guizhou province, China. The traffic data are collected every 30 seconds over 5 consecutive working days, dividing a day into 1680 time intervals. The original traffic flow data structure is shown in Table 1. The first four days' traffic data are used to construct the predictors, while the last day's data are used to test the performance of the predictor. In the following subsections, the spatiotemporal correlations with No. $tl_{25}, tl_{26}, tl_{27}, tl_{33}$, and tl_{42} sensors are used as an example to illustrate how to build and optimize the GMDH-based model to predict the future traffic flow at sensor tl_{26}. The sensors' locations are approximately shown in Figure 2.

Table 1: Traffic flow data structure

Current Sensor	Destination Sensor	Traffic Flow
tl9	tl8	[2,0,1,0,0,0,0,. . . ,5]
tl10	tl17	[0,2,3,1,0,2,1,. . . ,9]
. . .	. . .	. . .
tl9	tl2	[2,0,0,0,0,0,0,. . . ,0]
tl10	tl11	[0,0,0,1,0,0,0,. . . ,1]

Given the time interval T, the traffic flow is presented as a $24\times60/T$-dimensional vector. Traffic Flow represents the traffic flow from Current Sensor to Destination Sensor at different time interval.

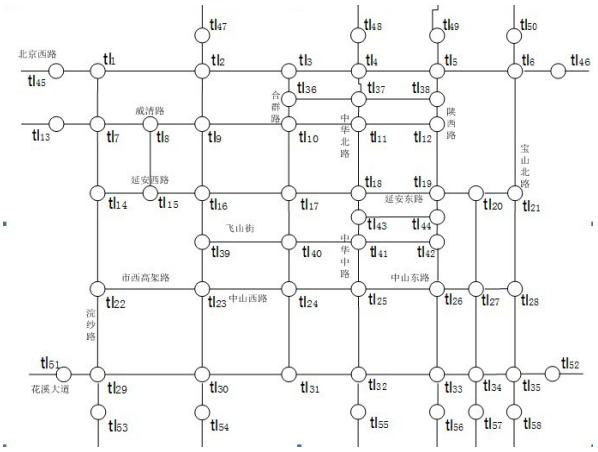


Figure 2: Nanming District of Guiyang

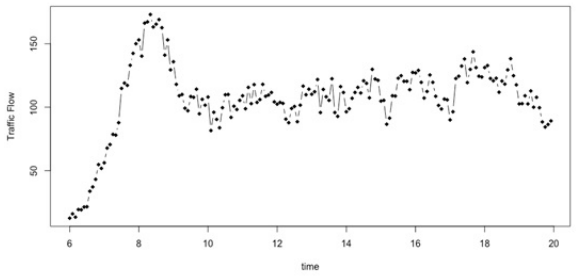
To examine the performance of the proposed approach with different time intervals, the data are aggregated into 5-, 10- and 20-minute time intervals according to [51]. In performing the aggregations, missing values were propagated upward. If the aggregated traffic flow series from a certain sensor remained unchanged for at least 1 hour, the succeeding observations with the same value are regarded as missing values.

4.2. Trend acquisition

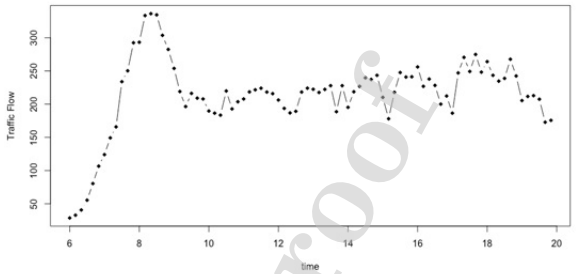
Recall that simple average trend over past W days is defined by Eqs. (2) and (3). The trend of the tl_{26} sensor over past 4 days with 5-minute interval is defined as a 168-dimension vector. Since the time interval is 5 minutes, each vector contains 168 observation each day.

$$\bar{Y}_{tl26} = \left[\frac{1}{4}\sum_{k=1}^{4} y_{tl26}^{k,1}, \frac{1}{4}\sum_{k=1}^{4} y_{tl26}^{k,2}, ..., \frac{1}{4}\sum_{k=1}^{4} y_{tl26}^{k,168} \right] \tag{14}$$

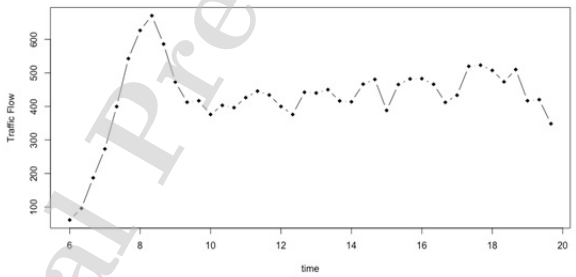
The trend of the tl_{26} sensor over past 4 days with 5-, 10- and 20-minute time intervals are defined as 168-, 84-, and 42-dimension vectors, respectively, and their graphical presentations are presented in Fig 3(a), (b), and (c), respectively.



(a) The trend of the tl_{26} sensor over past 4 days with 5-minute



(b) The trend of the tl_{26} sensor over past 4 days with 10-minute



(c) The trend of the tl_{26} sensor over past 4 days with 20-minute

Figure 3: The trend of the tl_{26} sensor over past 4 days with different time intervals

From Fig 3, one could see all the trends of tl_{26} sensor with different time intervals. From this figure, it is evident that intra-day traffic flow begins rising at 6:00, and then reaches its highest value at 8:00. During 9:00 to 18:00, the

traffic data fluctuate within a specific range and reache the second highest value at approximately 18:00, and then traffic decreases until 20:00. The phenomenon is called the morning peak and evening peak of traffic flow. All of these figures illustrate that traffic flows have a similar pattern over different days.

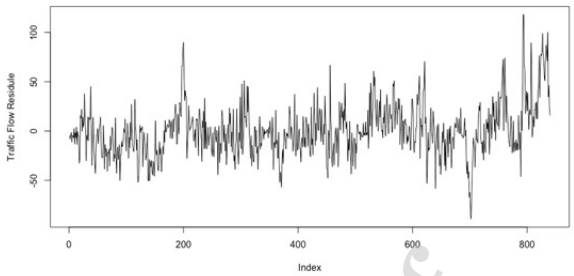
4.3. Residual Acquisition

Recall that traffic residual series is defined by Eqs. (4) and (5). We obtain 5 residual series for each sensor with the 5-minute interval by subtracting the simple average trend from the original traffic series, denoted as R_1, R_2, R_3, R_4, R_5. Each R_j is a cascaded residual series contains 4×168 observations, since the time interval is 5 minutes, and we have 168 observation each day. Then the $R_j^1, R_j^2, R_j^3, R_j^4$ residual series are cascaded consecutively. The cascaded residual series of the $tl_{25}, tl_{26}, tl_{27}, tl_{33}$, and tl_{42} sensors are fed into GMDH. R_{tl26}^5 is used to test the performance of predictor. The residual series of the tl_{26} sensor with different time intervals are presented in Figure 4 (a), (b), and (c). It is clear that the nonstationary elements of residual series are partially removed by detrending operation, as the mean and variance of residual are almost constant. The rest of nonstationary elements can be adjusted by the SARIMA and GARCH model in the next step. Note that the GMDH is used to model the conditional heteroscedasticity of the residual series only.

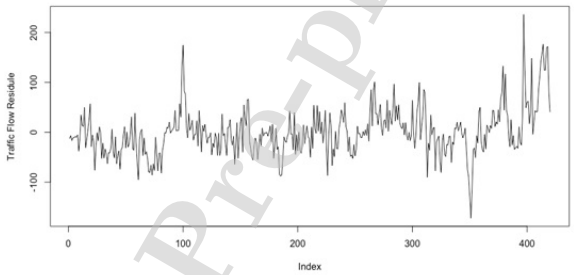
4.4. Predictor Architecture Settings

Regarding the structure of GMDH, we must determine the number of input nodes and the type of partial description. The number of hidden layers and the number of neruons in each hidden layer are determined by the self-organizing feature of GMDH. In order to determine the periods to lag of residual series from the $tl_{25}, tl_{26}, tl_{27}, tl_{33}, tl_{42}$ and tl_{25} sensors, the SARIMA and GARCH(1,1) models are used to fit these residual series. The optimal SARIMA models are selected based on the AIC criteria. Finally, the periods to lag (the parameter p of SARIMA) of five residual series from the $tl_{25}, tl_{26}, tl_{27}, tl_{33}$, and tl_{42} sensors over different time intervals are listed in table 2, and the number of input nodes

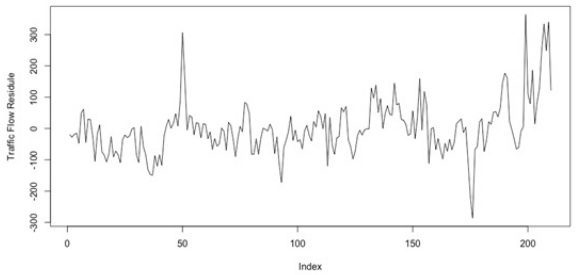
equals to the sum of lagged periods. Four types of reference functions, the linear, linear covariation, quadratic and cubic reference function are used as the partial description functions in this paper. For the detailed definitions, readers could refer to Eqs. (10)-(12).



(a) the residual series of the tl_{26} sensor with 5-minute time intervals



(b) the residual series of the tl_{26} sensor with 10-minute time intervals



(c) the residual series of the tl_{26} sensor with 20-minute time intervals

Figure 4: the residual series of the tl_{26} sensor with different time intervals

Table 2: the lagged variables (p) of 5 residual series from the tl_{25}, $tl_{26}, tl_{27}, tl_{33}$, and tl_{42} sensors over different time intervals

	tl_{26}	tl_{25}	tl_{27}	tl_{42}	tl_{33}
5-minute	6	4	4	4	5
10-minute	4	2	2	2	8
20-minute	3	2	6	3	3

4.5. Accuracy measures

In this empirical study three prevalent measures suggested by [52, 2], named magnitude of relative error (MRE), prediction accuracy at q (pred(q)), and root mean squared error (RMSE) are used to evaluate the performance of prediction models.

4.5.1. The magnitude of relative error (MRE)

The MRE is a normalized measure of the discrepancy between the predicted values $\hat{y}$ and real values y. It is defined as follows:

$$MRE = \frac{|y - \hat{y}|}{y} \tag{15}$$

4.5.2. Prediction accuracy at q, pred(q)

pred(q) is used to reflect how much the predicted MRE values are less than or equal to a specified percentage q. The value of pred(q) is in the range [0,1] and it is defined as follows:

$$pred(q) = \frac{K}{N} \tag{16}$$

where q is the specified percentage, K is the number of predictions whose MRE is less than or equal to q, and N is the total number of predictions. In this paper, pred(0.25) is used since this index is widely used in many traffic flow prediction related literatures. The larger pred(0.25) means the more predictions which MRE is less than or equal to 25%.

4.5.3. RMSE

Root Mean Square Error (RMSE) is the standard deviation of the prediction errors that presents how concentrated the data is around the prediction result. It is defined as follows:

$$RMSE = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{17}$$

5. EMPIRICAL RESULTS

The empirical results of proposed approach and baseline approaches are presented in this section.

5.1. Performance investigation

The purpose of this subsection is to show the performance of GMDH model with different partial description functions and time intervals. Here, we attempt to determine which partial description function has the best performance for what kind of time interval. In this paper, we have three choices of reference function: linear, linear covariation and cubic reference function. We can also combine them optionally. We choose seven types of combination (see RFC collumn in Table 3). Note that in the rest of the paper index 1, 2, and 3 are

adopted to denote the linear, linear covariation, and cubic reference functions, respectively.

5.1.1. Results

The comprehensive results on the datasets are presented in Table 3.

5.1.2. Analysis

Upon observing Table 3, it is clearly observed that the pred(0.25) of GMDH based predictor for 5 minutes is over 80%, for 10 and 20 minutes are around 76% and 4.8%, respectively, indicating that most predicated values for 5 and 10 minutes have a discrepancy with actual values less than 25%, except for 20 minutes. For 5-minute interval, the pred(0.25) are almost same for different types of combination. For 10-minute interval, cubic function can generate the better pred(0.25) than other types of combination. For 20-minute interval, the pred(0.25) values are not good for any type of combination. The reason can be explained as follows.

Table 3: The comprehensive results on the datasets

RFC	5min			10min			20min		
	RMSE	MRE	pred	RMSE	MRE	pred	RMSE	MRE	pred
1	22.2	0.15	0.82	47.2	0.15	0.786	81.362	15.011	0.048
1,2	23.1	0.16	0.83	54.2	0.16	0.774	76.823	14.961	0.048
1,2,3	22.9	0.15	0.82	55.3	0.17	0.726	77.591	12.434	0.048
1,3	22.6	0.15	0.82	798	0.44	0.702	74.016	13.614	0.071
2	24	0.16	0.82	60.6	0.17	0.798	80.382	15.761	0.048
2,3	23.1	0.16	0.82	195	0.21	0.75	78.14	13.051	0.048
3	23.5	0.16	0.83	48.7	0.15	0.845	79.688	15.578	0.024

Given the sampling time interval Δt and the highest frequency component in a time series f_{max}, the relationship between Δt and f_{max} is formalized by the Nyquist sampling theorem $\Delta t \leq \frac{1}{2f_{max}}$. When the interval Δt becomes larger, components with high frequency in the traffic residual flow series cannot be sampled and the information contained in such high frequency components is lost. It leads to the inaccurate prediction for 20-minute. From the MRE in

Table 3, we could see that the MREs of 5 and 10 minutes lie between 0.15 to 0.441, which means the GMDH based predictor have enough robustness. More precisely, it means that the discrepancy between actual values and predicted values is small. For 5-minute interval, MREs are still quite satisfactory for any type of combination. For 10-minute interval, cubic function can generate the best results than other combination. For 20-minute interval, the MRE are not good for any type of combination. Similarly, the RMSEs of the 5-minute interval are still encouraging, which is in the range of 22-24. However, we should notice that there are two abnormal results in 10-minute group, one using the linear combined with cubic function, and the other using linear covariation combined with cubic function. The results show that complex combinations do not necessarily produce the best prediction results. For 20-minute group, RMSEs are not good no matter the form of the reference function is. In summary, linear reference function can generate relatively better results for 5-minute interval, and cubic function can generate the best results for 10-minute time interval. For 20-minute interval, the GMDH cannot generate sufficiently good results, since the sampling time interval is too large, according to the Nyquist sampling theorem mentioned above, there are too much information contained in high frequency traffic residual series are lost.

5.2. Performance comparation

The purpose of this section is to find out what extent of prediction accuracy is improved by our approach comparing with LSTM based predictors. The linear combined with linear covariation function is used as the reference function for GMDH.

5.2.1. Results

The comprehensive results on the datasets are presented in Table 4.

5.2.2. Analysis

From Table 4, it is observed that comparing with LSTM, three measurements of GMDH are improved in varying degree for different time intervals. The results

Table 4: Performance comparation

Methods	5min			10min			20min		
	RMSE	MRE	pred(0.25)	RMSE	MRE	pred(0.25)	RMSE	MRE	pred(0.25)
GMDH	23.115	0.156	0.833	54.156	0.164	0.774	76.823	14.961	0.048
LSTM	39.178	0.183	0.76	163.532	0.576	0.036	490.305	0.959	0

show that the GMDH based predictor proposed improved both accuracy and robustness of prediction in traffic flow. For 5-minute interval, the improvements for RMSE, MRE and pred(0.25) are 16.06%, 2.7% and 7.3%. For 10-minute interval, improvements for RMSE, MRE and pred(0.25) are 109.38%, 41.2% and 73.8%. For 20-minute interval, it should be noticed that the results both for GMDH and LSTM are not so desirable as before. The reason has been mentioned in above section. However, the RMSE of GMDH based predictor is 74.016, which is still much better than that of the LSTM based predictor. Regarding MRE, however, the performance of the GMDH is worse than LSTM, which means that the robustness of the proposed approach, compared with the previous groups, is relatively weak while time interval becomes larger. For pred(0.25), although the two results are both not so good, the proposed approach are still better than the LSTM based predictor.

5.3. Interpretability Investigation

The structure of LSTM, the number of input nodes, the number of hidden layers, the number of neurons in each hidden layer, and the type of activation function must be predetermined. The optimal configuration of LSTM is always based on the trial and error pattern and the relationship between input and output of such model cannot be represented as an explicit form. So, the interpretability cannot be explained clearly. However, with the help of the self-organizing feature of GMDH, the casualty between the historical spatiotemporal information and the future traffic conditions is presented as an explicit form, a polynomial. In other words, the future traffic flow is interpreted as a polynomial function of the historical temporal and spatial traffic information. The

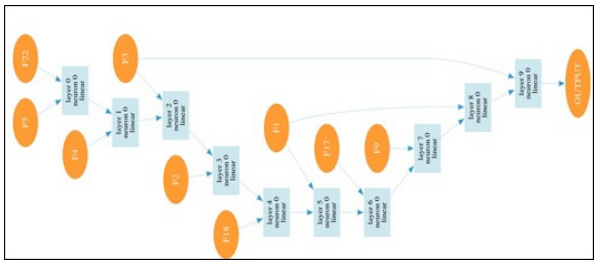
purpose of this section is to show what extent of interpretability is improved by the self-organizing feature of GMDH.

5.3.1. Results

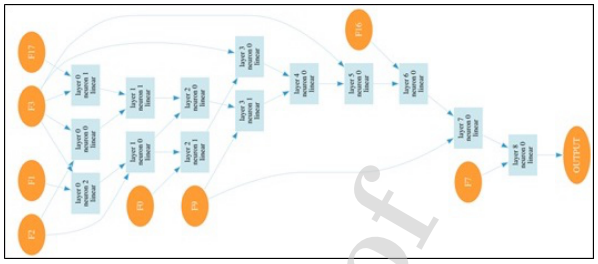
The structures of GMDH models for 5-minute, 10-minute and 20-minute are resented in Fig 5 (a) (b) and (c).

5.3.2. Analysis

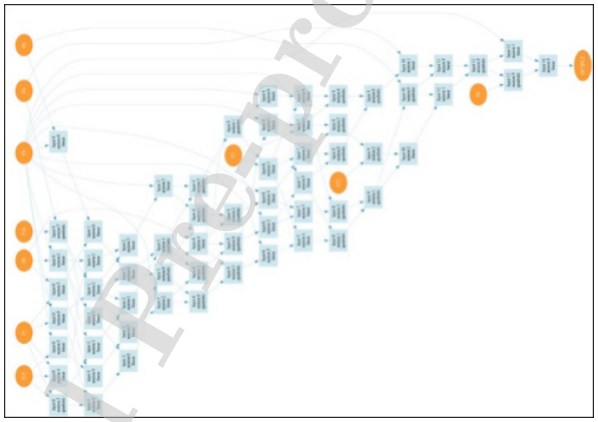
GMDH is a kind of self-organizing neural network, which means it can keep growing and the hidden layers are generated one by one instead of predetermining. On observing Fig 5(a), it is a 10-layer neural network with 23 input nodes while other 14 nodes are deleted, since the network believe these nodes have no spatiotemporal correlation with output. The causality between input and output variables of each neuron in the hidden layer can be explained by reference equation it adopted. For example, the neuron0 of layer0 has F22 and F5 two inputs nodes. The polynomial of neuron0 can be defined as $neuron0 = w_0 + w_1F_5 + w_2F_{22}$ and the output would be the input for the next layer with node F4. Finally, the polynomial of output node will be a complex polynomial which presents the spatiotemporal correlations of current traffic residual with the future traffic flow. Similarly, Fig 5(b) is a 9-layer neural network with 18 input nodes while other 10 nodes are deleted for the same reason mentioned above. The functions of every neuron in hidden layer are all linear, too. In Fig 5(c), the structure of neural network becomes more complex, and the number of neurons in hidden layers have increased. In addition, functions of some neurons are quadratic, which means full polynomial of the 2-nd degree, i.e. y^2. This is a 15-layer neural network with 15 input nodes. In summary, the self-organizing feature of GMDH leads to interpretability of the proposed method.



(a)



(b)



(c)

Figure 5: The structures of GMDH models for 5-minute, 10-minute and 20-minute

6. Conclusion

In this paper, we present an integrated approach for short term traffic flow prediction which combines GMDH with SARIMA together. It aims to simul-

taneously improve both the short term traffic flow prediction accuracy and the interpreting ability between the spatiotemporal information and future traffic condition. Extensive experiments show that the proposed model is effective and robust for different time intervals, and can significantly outperform the state-of-the-art prediction method, LSTM.

In the future, we plan to extend our experiments on different traffic scenarios to verify the generalization and validity of our approach.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61462092, 61402397, 61262024, 61573106 and 61379032) and the Yunnan Science and Technology Innovation Team Project (2017HC012).

References

[1] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias, Short-term traffic forecasting: Where we are and where we're going, Transportation Research Part C: Emerging Technologies 43 (2014) 3–19. doi:10.1016/j.trc.2014.01.005. URL http://dx.doi.org/10.1016/j.trc.2014.01.005

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-y. Wang, Traffic Flow Prediction With Big Data : A Deep Learning Approach, IEEE Transactions on Intelligent Transportation Systems PP (99) (2014) 1–9. doi:10.1109/TITS.2014.2345663.

[3] B. M. Williams, L. A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, Journal of transportation engineering 129 (6) (2003) 664–672.

[4] W. Huang, W. Jia, J. Guo, B. M. Williams, G. Shi, Y. Wei, J. Cao, Real-time prediction of seasonal heteroscedasticity in vehicular traffic flow series, IEEE Transactions on Intelligent Transportation Systems (99) (2017) 1–11.

[5] A. Stathopoulos, M. G. Karlaftis, A multivariate state space approach for urban traffic flow modeling and prediction, Transportation Research Part C: Emerging Technologies 11 (2) (2003) 121–135.

[6] G. Yu, J. Hu, C. Zhang, L. Zhuang, J. Song, Short-term traffic flow forecasting based on Markov chain model, in: 2003 IEEE Intelligent Vehicles Symposium, IEEE, 2003, pp. 208–212.

[7] I. Okutani, Y. J. Stephanedes, Dynamic prediction of traffic volume through Kalman filtering theory, Transportation Research Part B: Methodological 18 (1) (1984) 1–11.

[8] J. Whittaker, S. Garside, K. Lindveld, Tracking and predicting a network traffic process, International Journal of Forecasting 13 (1) (1997) 51–61.

[9] H. Sun, H. X. Liu, H. Xiao, R. R. He, B. Ran, Short term traffic forecasting using the local linear regression model, in: 82nd Annual Meeting of the Transportation Research Board, Washington, DC, 2003.

[10] Y. Y. Y. Zhang, Y. Y. Y. Zhang, A. Haghani, A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model, Transportation Research Part C: Emerging Technologies 43 (2014) 65–78. arXiv:arXiv:1607.03443v1, doi:10.1016/j.trc.2013.11.011. URL http://dx.doi.org/10.1016/j.trc.2013.11.011

[11] G. A. Davis, N. L. Nihan, Nonparametric regression and short-term freeway traffic forecasting, Journal of Transportation Engineering 117 (2) (1991) 178–188.

[12] T. Zhang, L. Hu, Z. Liu, Y. Zhang, Nonparametric regression for the short-term traffic flow forecasting, in: Mechanic Automation and Control Engineering (MACE), 2010 International Conference on, IEEE, 2010, pp. 2850–2853.

[13] B. Ghosh, B. Basu, M. O'Mahony, Bayesian time-series model for short-term traffic flow forecasting, Journal of transportation engineering 133 (3) (2007) 180–189.

[14] S. Sun, C. Zhang, G. Yu, A Bayesian network approach to traffic flow forecasting, IEEE Transactions on intelligent transportation systems 7 (1) (2006) 124–132.

[15] M. S. Dougherty, M. R. Cobbett, Short-term inter-urban traffic forecasts using neural networks, International journal of forecasting 13 (1) (1997) 21–31.

[16] 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 Transactions on Intelligent Transportation Systems 13 (2) (2012) 644–654.

[17] F. Jin, S. Sun, Neural network multitask learning for traffic flow forecasting, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1897–1901.

[18] E. I. Vlahogianni, M. G. Karlaftis, J. C. Golias, Short-term traffic forecasting: Where we are and where we're going, Transportation Research Part C: Emerging Technologies 43 (2014) 3–19.

[19] Y. Li, C. Shahabi, A brief overview of machine learning methods for short-term traffic forecasting and future directions, SIGSPATIAL Special 10 (1) (2018) 3–9.

[20] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, J. Liu, LSTM network: a deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems 11 (2) (2017) 68–75.

[21] G. Montavon, W. Samek, K. R. Müller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing: A Review

Journal 73 (2018) 1–15. doi:10.1016/j.dsp.2017.10.011.
URL https://doi.org/10.1016/j.dsp.2017.10.011

[22] J. Li, X. Chen, E. Hovy, D. Jurafsky, Visualizing and understanding neural models in nlp, arXiv preprint arXiv:1506.01066 (2015).

[23] M. S. Ahmed, A. R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, no. 722, 1979.

[24] J. Barros, M. Araujo, R. J. Rossetti, Short-term real-time traffic prediction methods: A survey, 2015 International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2015 (June) (2015) 132–139. doi:10.1109/MTITS.2015.7223248.

[25] C. Chen, J. Hu, Q. Meng, Y. Zhang, Short-time traffic flow prediction with arima-garch model, in: 2011 IEEE Intelligent vehicles symposium, IEEE, 2011, pp. 607–612.

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

[27] J. Guo, W. Huang, B. M. Williams, Integrated heteroscedasticity test for vehicular traffic condition series, Journal of Transportation Engineering 138 (9) (2012) 1161–1170.

[28] R. S. Tsay, Analysis of financial time series, Vol. 543, John Wiley & Sons, 2005.

[29] R. F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, Econometrica: Journal of the Econometric Society (1982) 987–1007.

[30] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of econometrics 31 (3) (1986) 307–327.

[31] D. B. Nelson, Conditional heteroskedasticity in asset returns: A new approach, Econometrica: Journal of the Econometric Society (1991) 347–370.

[32] K. Sohn, D. Kim, Statistical model for forecasting link travel time variability, Journal of Transportation Engineering 135 (7) (2009) 440–453.

[33] J. Guo, W. Huang, B. M. Williams, Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification, Transportation Research Part C: Emerging Technologies 43 (2014) 50–64.

[34] H. R. Kirby, S. M. Watson, M. S. Dougherty, Should we use neural networks or statistical models for short-term motorway traffic forecasting?, International Journal of Forecasting 13 (1) (1997) 43–50.

[35] N. G. Polson, V. O. Sokolov, Deep learning for short-term traffic flow prediction, Transportation Research Part C: Emerging Technologies 79 (2017) 1–17. arXiv:1604.04527, doi:10.1016/j.trc.2017.02.024.
URL http://dx.doi.org/10.1016/j.trc.2017.02.024

[36] L. Zhang, Q. Liu, W. Yang, N. Wei, D. Dong, An improved k-nearest neighbor model for short-term traffic flow prediction, Procedia-Social and Behavioral Sciences 96 (2013) 653–662.

[37] X. Gong, F. Wang, Three improvements on KNN-NPR for traffic flow forecasting, in: The IEEE 5th International Conference on Intelligent Transportation Systems, IEEE, 2002, pp. 736–740.

[38] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, L. D. Han, Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions, Expert systems with applications 36 (3) (2009) 6164–6173.

[39] L. Mihaylova, R. Boel, A. Hegyi, Freeway traffic estimation within particle filtering framework, Automatica 43 (2) (2007) 290–300.

[40] S. Sun, X. Xu, Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction, IEEE Transactions on Intelligent Transportation Systems 12 (2) (2011) 466–475.

[41] E. Bolshinsky, R. Freidman, Traffic Flow Forecast Survey, Technion–Israel Institute of Technology.–2012.–Technical Report.–15 (2012) 1–15.
URL http://nwwwn.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2012/CS/CS-2012-06.pdf

[42] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17 (4) (2017) 818.

[43] H. Yu, Z. Wu, S. Wang, Y. Wang, X. Ma, Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks, Sensors 17 (7) (2017) 1501.

[44] X. Dai, R. Fu, Y. Lin, L. Li, F.-y. Wang, DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction (2017). arXiv:1707.03213.
URL http://arxiv.org/abs/1707.03213

[45] J. J. Schmidhuber, Deep Learning in Neural Networks: An Overview, Neural networks 61 (2015) 85—-117. arXiv:1404.7828, doi:10.18388/abp.2015_1002.

[46] C. Chen, W. Yin, L. Li, J. Hu, Z. Zuo, The retrieval of intra-day trend and its influence on traffic prediction, Transportation Research Part C Emerging Technologies 22 (5) (2012) 103–118.

[47] Z. Li, Y. Li, L. Li, A comparison of detrending models and multi-regime models for traffic flow prediction, IEEE Intelligent Transportation Systems Magazine 6 (4) (2014) 34–44. doi:10.1109/MITS.2014.2332591.

[48] L. Li, X. Su, Y. Wang, Y. Lin, Z. Li, Y. Li, Robust causal dependence mining in big data network and its application to traffic flow predictions, Transportation Research Part C: Emerging Technologies 58 (2015) 292–307.

[49] J. Guo, B. M. Williams, Real-time short-term traffic speed level forecasting and uncertainty quantification using layered kalman filters, Transportation Research Record 2175 (1) (2010) 28–37.

[50] L. Mo, L. Xie, X. Jiang, G. Teng, L. Xu, J. Xiao, GMDH-based hybrid model for container throughput forecasting: Selective combination forecasting in nonlinear subseries, Applied Soft Computing Journal 62 (December) (2018) 478–490. doi:10.1016/j.asoc.2017.10.033.
URL http://dx.doi.org/10.1016/j.asoc.2017.10.033

[51] L. C. Edie, Discussion of traffic stream measurements and definitions, Port of New York Authority, 1963.

[52] L. Qu, L. Li, Y. Zhang, J. Hu, PPCA-Based Missing Data Imputation for Traffic Flow Volume : A Systematical Approach 10 (3) (2009) 512–522. doi:10.1109/TITS.2009.2026312.