Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 8245850, 16 pages
https://doi.org/10.1155/2022/8245850

Research Article

A Combined Discrete Road Traffic State Prediction Model Based on GFD-ARMA-FISHER Analytical Framework

Pengcheng Yuan and Yin Han

Business School, University of Shanghai for Science and Technology, Shanghai 200093, China

Correspondence should be addressed to Pengcheng Yuan; danis_cx@126.com

Received 2 June 2022; Revised 28 October 2022; Accepted 31 October 2022; Published 1 December 2022

Academic Editor: Mahmoud Mesbah

With the popularization of the Internet and the widespread application of mobile terminal, travelers are increasingly dependent on traffic information. It is particularly important to construct and develop more accurate discrete traffic state prediction models. Considering that most of the previous studies about traffic state prediction are essentially a prediction of specific parameters, this paper proposes a generalized fractional difference (GFD)-auto regressive moving average (ARMA)-Fisher combined model that can directly predict the discrete traffic state. First, using the original historical traffic flow data as training samples for cluster analysis, we obtain different traffic state classification samples. Then, using the Fisher discriminant method, we develop a discrete traffic state recognition model based on the traffic state classification samples. The model can help us identify the discrete state of the future traffic flow when we input a set of predicted traffic parameters to it. In order to improve the accuracy of the model prediction, this paper creatively proposes to apply the GFD method to the stabilization of original traffic flow data and then develop an ARMA model to predict the traffic parameters processed by GFD (GFD-ARMA). Last, we obtain the predicted discrete traffic state using the calibrated Fisher discrimination model whose inputs are the parameters predicted by the GFD-ARMA. The proposed method is tested using field data from CHangzhou, China. The results suggest that the developed GFD-ARMA-FISHER method shows a higher accuracy for traffic state prediction and is better than other similar methods.

1. Introduction

With the rapid development of the social economy and the constant expansion of the size of cities, people's demands regarding the quality and quantity of transportation have been increasingly growing and, developing a fast and green traffic system has become the main focus of current research. Facing increasingly severe traffic congestion pressure, various countries around the world have adopted a variety of countermeasures, the most important of which is the development of intelligent transportation systems (ITSs) [1, 2]. Traffic flow parameter prediction, as the basis of an ITS, is the prerequisite and basis for traffic flow control and induced management [2–4]. Traffic flow parameter prediction refers to the estimation of traffic parameters in the future based on historical survey data. At present, the main methods used for traffic parameter prediction can be divided into three categories: nonparametric prediction methods, parameter prediction methods, and combinations of these methods.

Nonparametric prediction methods are new methods that have gradually been widely studied due to the rapid development of artificial intelligence and big data in recent years. These methods borrow the basic idea of artificial intelligence and mainly achieve predictions by training a model using observation samples. Nonparametric prediction methods mainly include the nearest neighbor method [5–7], neural networks [8, 9], the support vector machine [10–13], machine learning [14, 15], grey theory [16–18], and simulation methods [19]. Parameter prediction methods are based on the potential change in law in historical traffic data. The development of a parametric prediction model requires developing a parametric logical model, estimating the unknown parameters of the model, and testing the reliability of the unknown parameters. Parametric prediction methods

mainly involve the historical averaging method, the smoothing algorithm, the Bayesian combination method [20], the filtering method [21], time series prediction [22], and spectral analysis [23, 24].

The most classic and widely used parameter prediction method is the autoregressive linear analysis method, which mainly includes the auto regression (AR) model, the moving average (MA) model, the auto regression moving average (ARMA) model [25–27], the auto regression integrated moving average (ARIMA) model [9, 28], ARIMAX [29], the auto regression conditional heteroscedasticity (ARCH) model, the generalized auto regression conditional heteroscedasticity (GARCH) model [30, 31], and the combination of these models [32–35]. The ARMA and ARCH family models are based on stationary time series. It means that the ARMA model and ARCH model, which are traditional time series analysis tools, are all based "fully or almost independently" between two observations that are far apart in a time series. A stationary time series is a short memory time series with an autocorrelation coefficient decaying at an exponential rate. If a time series is not stable, the series should be stabilized first, and then it is possible to perform ARMA/ARCH modeling on it. In general, the most common method for stabilizing a time series is to calculate integer order differences. However, although integer order differences can stabilize nonstationary time series, the integer difference process also makes the original time series lose useful information, and this information may be able to help develop more accurate models [36]. Therefore, how to stabilize the original data with as much useful information as possible is a core issue studied in this paper. This paper considers introducing the generalized fractional difference (GFD) method into the prediction process of urban road traffic states.

Moreover, although there are many studies on "traffic state" estimation or prediction, the "traffic state" mentioned in these studies is actually a generalized concept characterized by various traffic parameters, including the volume [37], travel time, speed [38], capacity [39], density, cumulative number of vehicles [40], and so on. Therefore, "traffic state" prediction is actually the prediction of these parameters. These prediction parameters are useful and important for determining the degree of traffic congestion, but they are not intuitive and easily accepted parameters for travelers especially for people who are used to relying on traffic guidance information. Therefore, in this paper, we propose a method that can directly predict the discrete traffic state that reflects the degree of traffic congestion. In other words, the traffic state mentioned in this paper is a narrow-sense traffic state, which is characterized by smooth flows, general congestion, severe congestion, etc. The method proposed in this paper can directly use historical traffic parameters to make predictions about future discrete traffic congestion states. Following these ideas, first, this paper extracts the traffic flow time series parameters for a day on a fixed section of a road. Then, a traffic flow parameter prediction model based on the GFD-ARMA is developed. Next, using the historical data as training samples, a traffic state recognition model based on the Fisher discriminant is developed. The predicted traffic state can finally be obtained by organically combining these two models.

The highlights of this paper are as follows: First, this paper creatively proposes to apply the GFD method to the stabilization of original traffic flow data which can not only stabilize the original data but also make the stabilized data retain more effective information as much as possible. This practice provides a prerequisite for developing a more accurate prediction model. Second, different from the goal of only predicting traffic parameters in the past, this paper proposes a method that can directly predict discrete traffic states which is easy to adopt in practice. Finally, the new traffic state prediction method proposed in this paper combines the ideas of cluster analysis and discriminant analysis and cleverly uses the results of the cluster analysis as training samples for the calibration process of the discriminant analysis model, making full use of the effective information contained in the original data.

The rest of this paper is organized as follows. Section 2 mainly describes the GFD-ARMA-FISHER analytical framework. Section 3 introduces the data used in this paper; Section 4 gives the calibration process of the GFD-ARMA-FISHER combined model and the prediction results of the model; and finally, Section 5 gives the main conclusions and discussion.

2. GFD-ARMA-FISHER Analytical Framework

2.1. GFD-ARMA

2.1.1. The Model. For a time series $\{y_t\}$, define $\nabla_k y_t$ as the kth step (k is a positive integer parameter) difference of y_t. It can be calculated as follows:

$$\nabla_k y_t = y_t - y_{t-k}. \tag{1}$$

Let L be the delay operator. Then, for the time series $\{y_t\}$, define $y_{t-k} = L^k y_t$. Then, we can obtain the following equation:

$$\nabla_k y_t = \left(1 - L^k\right) y_t. \tag{2}$$

Define $\nabla^d y_t$ as the d-order difference of y_t. When d is a positive integer parameter greater than or equal to 1, it can be calculated as follows:

$$\nabla^d y_t = \nabla^{d-1} y_t - \nabla^{d-1} y_{t-1}. \tag{3}$$

When $d = 1$, $\nabla y_t = y_t - y_{t-1}$.

It is clear that $\nabla^d y_t$ can be written equivalently as follows:

$$\nabla^d y_t = (1 - L)^d y_t. \tag{4}$$

According to the polynomial expansion rule, $(1 - L)^d$ can also be written as follows:

$$(1-L)^d = 1 - dL + \frac{d(d-1)}{2!}L^2 - \frac{d(d-1)(d-2)}{3!}L^3 + \cdots + (-1)^i \frac{d(d-1)(d-2)\cdots(d-\mathrm{i}+1)}{i!}L^i + \cdots + (-1)^d L^d,$$
$$= \sum_{i=0}^{d} (-1)^i \frac{d!}{i!(d-i)!}L^i. \tag{5}$$

We refer to the above methods (equations (4) and (5)) to calculate the difference in a time series as the integer difference method. However, when d is not just a positive integer but is also a general positive fractional parameter, $(1-L)^d$ can be written as follows:

$$(1-L)^d = 1 - dL + \frac{d(d-1)}{2!}L^2 - \frac{d(d-1)(d-2)}{3!}L^3 + \cdots + (-1)^i \frac{d(d-1)(d-2)\cdots(d-\mathrm{i}+1)}{i!}L^i$$
$$+ \cdots + (-1)^k \frac{d(d-1)(d-2)\cdots(d-k+1)}{k!}L^k = \sum_{i=0}^{k} (-1)^i \frac{d(d-1)(d-2)\cdots(d-k+1)}{k!}, \tag{6}$$

where k is a preset positive integer and represents the memory length of the time series. We refer to the above method (equations (4) and (6)) to calculate the difference in a time series as the generalized fractional difference (GFD) method.

We state that a nonstationary time series $\{y_t\}$ can be fitted by the GFD-ARMA model if it can be fitted by the ARMA after it is stabilized by the GFD. The whole GFD-ARMA model can be expressed as follows:

$$\varphi(L)(1-L)^d y_t = \theta(L)\varepsilon_t + c. \tag{7}$$

ε_t is a white noise series $(E(\varepsilon_t) = 0, Var(\varepsilon_t^2) = \sigma^2, E(\varepsilon_t\varepsilon_s) = 0\, s \neq t)$, and c is a constant. In the actual modeling process, whether the model contains c depends on whether c is a constant. In the actual modeling process, whether c is included in the model depends on the needs of the model (especially the impact on the goodness of fit of the model). $\varphi(L) = 1 - (\varphi_1 L + \varphi_2 L^2 + \cdots + \varphi_p L^p)$, p is the lag order of the auto regression, φ_i $(i = 1, 2, \cdots, p)$ is the parameter of the lag term y_{t-i}, $\theta(L) = 1 - (\theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q)$, q is the lag order of the moving average, and θ_j $(j = 1, 2, \cdots, q)$ is the parameter of the moving average lag term ε_{t-j}.

The unknown parameters that need to be estimated in Model (7) include $d\varphi_i(i = 1, 2, \cdots, p)\theta_j$ $(j = 1, 2, \cdots, q)$. However, before estimating the parameters, we must first determine the lag term of the model. In this paper, we first estimate parameter d, then determine the lag term of the model, and finally, estimate parameters φ_i $(i = 1, 2, \cdots, p)$ and θ_j $(j = 1, 2, \cdots, q)$. These parameters can be estimated as follows.

2.1.2. Parameter Estimation. Estimate parameter d.

First: Initialize $d = 0$, and set a small step size δ, such as $\delta = 0.1$.

Second: Test the stationarity of the sequence $\{(1-L)^d y_t\}$ (for a time series $\{y_t\}$, differentiate it using the GFD method (equations (4) and (6)). If it is stable, then $d = 0$ and end the process; otherwise, let $d_{new} = d + \delta$.

Third: Test the stationarity of the new sequence $\{(1-L)^{d_{new}} y_t\}$. If it is stable, then $d = d_{new}$ and end the process; otherwise, let $d_{new} = d_{new} + \delta$ and repeat the third step.

Determination of the lag terms of the model: Generally speaking, in practice, it is difficult to determine the lag terms of the model accurately due to the complexity of the actual data. Although the Box-Jenkins method [41] is used to estimate the lag terms in many studies, this paper believes that the Box-Jenkins method error may increase for data with greater noise. Therefore, based on the Box-Jenkins method, this paper proposes a method to determine the lag terms, which is suitable for the data in this paper. The process is as follows.

First: For a stationary time series $\{y_t\}$, calculate its autocorrelation coefficient (AC) and partial autocorrelation coefficient (PAC) and test its significance. Let the set of all terms with significant AC parameters be $P_m = \{\cdots, p_i, \cdots, p_j, \cdots\}$ (there are m terms that are significant) and the set of all terms with significant PAC parameters be $Q_n = \{\cdots, q_i, \cdots, q_j, \cdots\}$ (there are n terms that are significant).

Second: Develop sparse ARMA$(\overline{P}_m, \overline{Q}_n)$ models, where $\overline{P}_m$ and $\overline{Q}_n$ are subsets of P_m and Q_n, respectively.

Third: Test the residual sequences of these models. Discard the model if its residual series are correlated.

Fourth: Compare the fit goodness of the remaining models. The model with the best fit will be selected as the final model. If the two models have an equal goodness of fit, then choose the model with the smaller AIC value. Then, p and q can be determined.

The current estimation methods for parameters φ_i and θ_j are relatively mature, and the related details of the estimation methods can be found in reference [42, 43]. This paper estimates these parameters using the Eviews10.

2.2. Fisher Discriminant Analysis Method. The Fisher discriminant analysis method is a supervised classification method that is based on the idea of projection (or dimensional reduction). The Fisher discriminant analysis method uses a few linear combinations of p variables $(x_1, x_2, \cdots, x_p)$ (called discriminants), where $y_1 = a_1^{'}\mathbf{x}, y_2 = a_2^{'}\mathbf{x}, \cdots, y_r = a_r^{'}\mathbf{x}$ (generally, r is significantly smaller than p), to replace the original p variables $(x_1, x_2, \cdots, x_p)$ to achieve dimensionality reduction. Moreover, using the r discriminants $y_1, y_2, \cdots, y_r$, the assignment of samples is discriminated. It is supposed that the p-dimensional observed values of group π_i are x_{ij}, where $j = 1, 2, \cdots n_i$ and $i = 1, 2, \cdots k$, which are projected to a p-dimensional constant vector a. This process obtains the projection points that separately correspond to the linear combinations $y_{ij} = a^{'}x_{ij}$, where $j = 1, 2, \cdots n_i$ and $i = 1, 2, \cdots k$. In this way, all the p-dimensional observed values are simplified into one-dimensional observed values. Let $\overline{y}_i$ denote the mean of y_i in group π_i and $\overline{y}$ stand for the mean of all $y_{ij}s$ in group k; that is,

$$\overline{y}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} y_{ij} = \mathbf{a}'\overline{\mathbf{x}}_i,$$
$$\overline{y} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i} y_{ij} = \mathbf{a}'\overline{x}, \tag{8}$$

where $n = \sum_{i=1}^{k} n_i$, $\overline{\mathbf{x}}_i = 1/n_i \sum_{j=1}^{n_i} x_{ij}$, and $\overline{\mathbf{x}} = 1/n \sum_{i=1}^{k} n_i \overline{\mathbf{x}}_i$. The specific process of Fisher discriminant analysis is as follows.

Step 1: Calculate the sum of squares and the cross product sums of y_{ij} between groups: $B = \sum_{i=1}^{k} n_i (\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})^{'}$.

Step 2: Calculate the sum of squares and cross product sums of y_{ij} within a group: $E = \sum_{i=1}^{k}\sum_{j=1}^{n_i}(x_{ij} - \overline{x}_i)(x_{ij} - \overline{x}_i)^{'}$. (if $\sum_{i=1}^{k} n_i - k \geq p$).

Step 3: Calculate the total nonzero eigenvalues of $E^{-1}B$, which are separately displayed as $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_s > 0$ and separately correspond to the eigenvectors $t_1, t_2, \cdots, t_s$. Therefore, s discriminants ($y_1 = t_1^{'}x$, $y_2 = t_2^{'}x \cdots\cdots$ $y_s = t_s^{'}x$) can be acquired.

Step 4: Calculate the group mean of the discriminant $\overline{y}_{ij}$.

Step 5: Calculate the cumulative contribution $\sum_{i=1}^{r}\lambda_i / \sum_{i=1}^{r}\lambda_i$ of the first r discriminants. If the cumulative contribution of the first r discriminants reaches a high proportion, then the first r discriminants can be used for discrimination. The discrimination rule is shown as follows:

$$\mathbf{x} \in \pi_l \text{if} \sum_{j=1}^{r}\left(y_j - \overline{y}_{lj}\right)^2 = \min_{1\leq i\leq k}\sum_{j=1}^{r}\left(y_j - \overline{y}_{ij}\right)^2. \tag{9}$$

2.3. The Combined Analytical Framework. The discrete traffic state prediction process studied in this paper is shown in Figure 1. First, this paper extracts historic traffic flow time series parameters to develop the traffic parameter prediction model using the GFD-RAMA method. In addition, the traffic state training samples are obtained by the k-means cluster method. Then, using the traffic state training samples, we develop the traffic state recognition model using the Fisher discriminant analysis method. Last, the new traffic state can be predicted via the comprehensive application of these two models.

3. Data Preparation and Preprocessing

3.1. Traffic Parameter Data. The data used in this paper are from a section of Bo'ai road in the center of Changzhou city from 7/10/2008~9/10/2008. The data include four types of traffic parameters (average quantity, average speed, average occupancy and average headway). These four types of parameters are denoted as $Quantity_t$, Speed_t, Occupancy_t and Headway_t, respectively, in this paper. The data are divided into two types: one type is used to calibrate the model, and the other type is used to validate the model. The model is calibrated with data from 7/10/2008 (Tuesday, a workday), and the model is validated with data from 8/10/2008 and 9/10/2008. The descriptive statistics of all the data are shown in Table 1.

3.2. Traffic State Sample. The premise of developing a traffic state recognition model is to have a large number of traffic state classification samples. However, it is well known that, in practice, although traffic flow parameter data are usually easy to obtain, traffic state data are usually not easy to obtain for the following reasons. First, the traffic state parameter is not a parameter that can be directly collected. The traffic state parameter is secondary data after processing the original traffic flow parameters. In addition, different cities and regions have different standards for the definition and classification of traffic states (for example, some cities use speed as the only criterion for classifying traffic conditions, and some cities use manual judgment results as judgment criteria). Second, because different roads have different attributes and different travelers have different perceptions of traffic conditions, it is almost impossible to construct a unified and absolute classification standard for traffic states. Although an absolute traffic state classification is not easy to obtain, we can still obtain relative traffic state classifications using some methods (such as classification methods based on unsupervised learning). This paper applies K-means clustering to unsupervised learning and the classification of traffic states. The purpose is to obtain classification samples, lay the foundation for developing the recognition model, and verify the accuracy of the state prediction model. In our opinion, this classification method is more general and objective. The classification of the traffic states of a road by the model can indicate the relative congestion of the road. Considering that this paper mainly hopes to use classification samples to build a traffic state prediction model and discuss the accuracy of the state prediction model, it divides the relative traffic state into 3 classes and 5 classes (that is, for K-means clustering, set the number of classifications to 3 and 5, respectively, in advance). The accuracy of the model prediction results under different

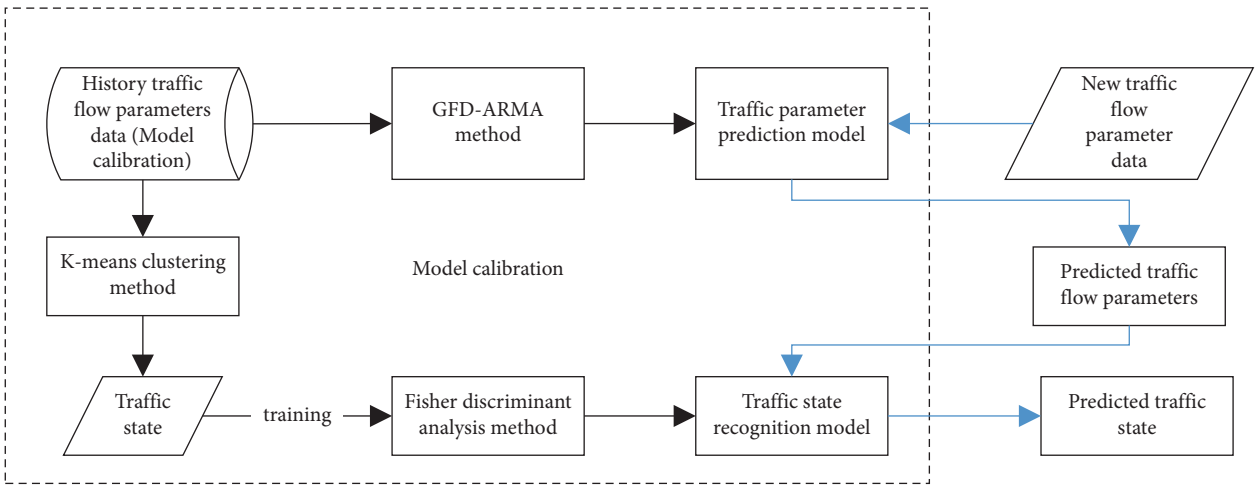

Figure 1: Traffic state prediction process.

Table 1: Descriptive statistics of sample data.

Data type	Sample size	Data for model calibration				Data for model testing							
		Date: 7/10/2008				Date: 8/10/2008				Date: 9/10/2008			
		Min	Max	Mean	Sd	Min	Max	Mean	Sd	Min	Max	Mean	Sd
Quantity	480	0	78	34.4	22.0	0	76	34.0	21.6	0	77	34.6	22.1
Speed	480	34	70	46.0	5.6	21	70	45.6	6.3	31	70	45.6	5.5
Occupancy	480	0	18.3	6.9	5.0	0	45.44	7.0	5.7	0	27.91	6.9	5.3
Headway	480	4.73	180	33.51	44.96	4.95	180	33.27	44.52	5.62	179	32.96	44.25

numbers of classifications is discussed separately as well. According to the clustering process below, we can obtain the classification results of the traffic state.

Step 1: Determine the number of classes that need to be clustered, that is, the number of clusters.

Step 2: Initialize the cluster center. Choose k data randomly as the initial k classes. Obviously, each datum is also the cluster center of each class.

Step 3: For each datum, calculate its distance from each cluster center (Euclidean distance). A datum will be divided into a certain class if it is closest to the cluster center of this class. Then, k new classes are generated.

Step 4: Calculate the mean value of each new class as the new cluster center.

Step 5: Terminate the process. If the distance between the new clustering center and the original clustering center is less than a certain threshold, then terminate the process; otherwise, repeat Steps 3 to 5.

Preset the number of categories to 3 and 5, respectively (clustering the traffic congestion status into 3 and 5 categories, respectively), using the K-Means clustering method, we can get the classification results of the road traffic states samples, Table 2 shows the cluster centers of three states and five states.

As shown in Table 2, State 2 has the smallest average Quantity, the largest average Speed, and the smallest average occupancy compared to the other states. State 3 has the largest average Quantity, the smallest average Speed, and the largest average occupancy compared to the other states. State 1 lies between states 1 and 3. Therefore, state 2 can be interpreted as the smoothest state, state 3 is the most crowded state compared to state 2, and state 1 is between state 2 and state 3. According to the same reason, we can arrange the states in Table 2 from smooth to crowded in the following order: state 5, state 3, state 1, state 2, and state 4. This shows that the state clustering obtained by the K-means clustering method can be used as a method of discrete traffic state classification (Table 3).

3.3. Stabilize the Data. The stability test is the premise of developing a stationary time series model, which can effectively avoid a spurious regression. This paper uses the ADF test method to test the stationarity of the traffic parameters. The results are shown in Table 4.

As seen from the above table, the T statistics of the ADF test of the four types of parameters are all greater than the T values at the significance levels of 1%, 5%, and 10%, which means that the four types of parameters are unstable. Therefore, the parameters should be stabilized before modeling and analysis.

According to the process given in 2.1.2 the parameter d can be estimated. At a significance level of $p = 0.05$, the estimated value of d can be obtained ($d_{\text{Quantity}} \approx 0.7$, $d_{Speed} \approx 0.6$,$d_{\text{Occupancy}} \approx 0.7$, and $d_{\text{Headway}} \approx 0.7$). The data processed by the GFD method are shown in Figure 2. (The parameters processed by the GFD method are denoted as Quantity$f$, Speed$f$, Occupancy$f$, and $Headway f$).

Table 2: Cluster centers of every state.

	3 states			5 states				
State	1	2	3	1	2	3	4	5
Quantity	38	8	58	39	51	23	64	6
Speed	45	50	42	46	43	48	40	50
Occupancy	7.10	1.18	12.87	7.23	10.7	3.58	14.75	0.82

Table 3: ADF test results.

	Trend and intercept		Intercept		None	
	t-statistic	Prob.*	t-statistic	Prob.*	t-statistic	Prob.*
Quantity	−0.839	0.960	−1.393	0.586	−0.732	0.398
Speed	−2.661	0.253	−2.764	0.064	−0.344	0.560
Occupancy	−0.366	0.988	−1.196	0.677	−0.695	0.415
	−3.977	1%	−3.443	1%	−2.569	1%
Test critical values	−3.419	5%	−2.867	5%	−1.941	5%
	−3.132	10%	−2.569	10%	−1.616	10%

The stability test results are shown in Table 5. As seen from Table 4, after GFD processing, at the 1% significance level, all types of parameters passed the ADF test.

4. Model Calibration and Prediction Results

4.1. Estimate Parameters $\varphi(L)$ and $\theta(L)$. Using the estimation method described in 2.1.2, after a large number of model comparisons and analyses, we finally determine the lag terms of the three types of parameters and then estimate the parameters $\varphi(L)$ and $\theta(L)$. The results are shown in Table 6.

It can be seen from Table 6 that the AR lag terms of the ARMA model of the traffic flow parameter Quantity$f$ are the 1st, 2nd, 3rd, and 4th terms, and that the MA lag terms are the first and second terms, respectively. Using the lag term as the explanatory variable to develop the parameter-predicted ARMA model, each item is significant at the 0.05 significance level according to t-tests (the corresponding significance probabilities of the terms are all less than 0.05). In the same way, it can be seen that the AR lag terms and the MA lag term of Speed$f$ are the 1st, 9th, 10th, and 12th terms and the 1st term, respectively, and that the AR lag terms and the MA lag term of Occupancy$f$ are the 1st, 2th, 3th, 6th, 7th, and 8th terms and the 1st term, respectively. To test the effect of the developed model, we performed a serial correlation test on the residuals of the model. The Q_{LB} test results of the residual sequence correlation are shown in Figure 3. As seen from Figure 3, the residual sequences of the three types of parameters all show sequence independence (the Prob values of the Q_{LB} statistics are all greater than 0.05), so it can be considered that the developed model can fully extract the regular information included in the original data. The data fitted by the model (Table 5) are compared with the original data, as shown in Figure 4. It can be seen from Figure 4 that the law of the data after fitting are basically consistent with the original data, which shows that the model can better fit the original data.

4.2. Parameter Prediction Model. Using the collected historical data (such as Quantity_{t-1},Quantity_{t-1},$\cdots$, *and*Quantity_1), the GFD series (including Quantityf_{t-1} ,Quantityf_{t-1},$\cdots$, and Quantityf_{t-k}), and the calibrated model (shown in Table 4), the flow parameters at the future time t can be predicted. The prediction model is expressed as follows:

$$\begin{cases} \widehat{\text{Quantity}}_t = \widehat{\text{Quantity}}f_t + \sum_{i=1}^{k} (-1)^{i+1} \dfrac{d(d-1)(d-2)\cdots(d-k+1)}{k!} \text{Quantity}_{t-i}, \\ \widehat{\text{Speed}}y_t = \widehat{\text{Speed}}f_t + \sum_{i=1}^{k} (-1)^{i+1} \dfrac{d(d-1)(d-2)\cdots(d-k+1)}{k!} \text{Speed}_{t-i}, \\ \widehat{\text{Occupancy}}_t = \widehat{\text{Occupancy}}_t + \sum_{i=1}^{k} (-1)^{i+1} \dfrac{d(d-1)(d-2)\cdots(d-k+1)}{k!} \text{Occupancy}_{t-i}, \end{cases} \tag{10}$$

where

Table 4: ADF test results (the data are processed by the GFD method).

	Trend and intercept		Intercept		None	
	t-statistic	Prob.*	t-statistic	Prob.*	t-statistic	Prob.*
Quantity f	−4.770	0.0006	−4.645	0.0001	—	—
Speed f	−16.419	0.0000	−16.349	0.0001	—	—
Occupancy f	−4.832	0.0005	−4.776	0.0001	—	—
	−3.977	1%	−3.443	1%	−2.569	1%
Test critical values	−3.419	5%	−2.867	5%	−1.941	5%
	−3.132	10%	−2.569	10%	−1.616	10%

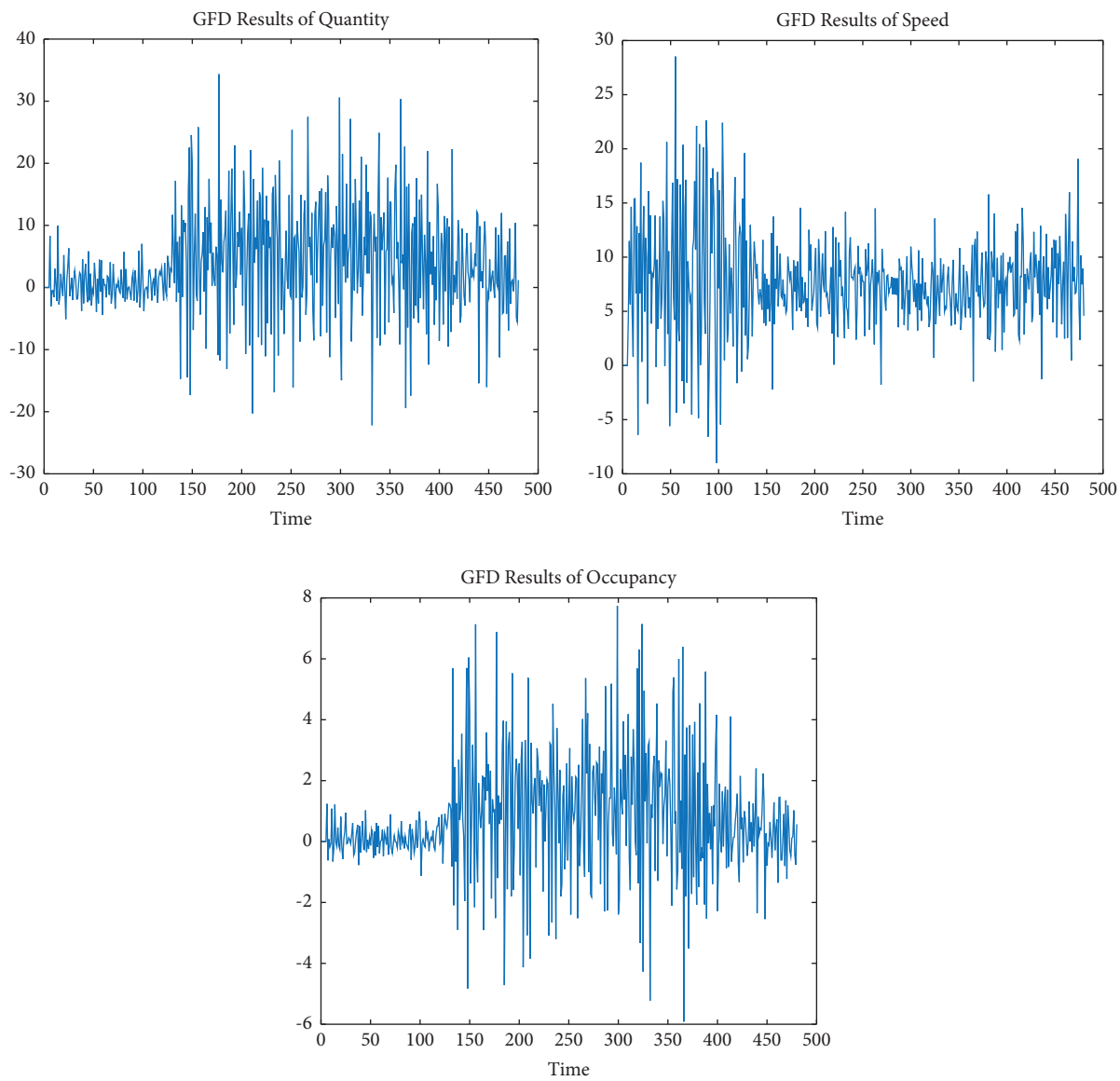

Figure 2: Data processed by the GFD.

Table 5: Lag terms and parameter estimation results.

Lag terms and parameter estimation results of Quantity f								
Lags		Constant	AR (p)					MA (q)
		C	1	2	3	4	1	2
Parameter estimation results		—	$\widehat{\varphi}_1$	$\widehat{\varphi}_2$	$\widehat{\varphi}_3$	$\widehat{\varphi}_4$	$\widehat{\theta}_1$	$\widehat{\theta}_2$
			1.172	−0.150	0.158	−0.182	−1.728	0.753
Significance test	t-statistic	—	9.125	−2.057	2.547	−2.267	−14.467	6.803
	Prob (t-statistic)	—	<0.001	0.040	0.011	0.023	<0.001	17.646
R-squared						0.303		
Adjusted R-squared						0.295		
Lag terms and parameter estimation results of Speed f								
Lags		Constant	AR (p)				MA (q)	
		C	1	9	10	12	1	
Parameter estimation results		7.599	$\widehat{\varphi}_1$	$\widehat{\varphi}_9$	$\widehat{\varphi}_{10}$	$\widehat{\varphi}_{12}$	$\widehat{\theta}_1$	
			0.387	0.117	0.062	0.131	−0.699	
Significance test	t-statistic	6.016	3.757	2.160	3.897	−13.194	−14.467	
	Prob (t-statistic)	<0.001	<0.001	0.031	<0.001	<0.001	<0.001	
R-squared						0.117		
Adjusted R-squared						0.106		
Lag terms and parameter estimation results of Occupancy f								
Lags		Constant	AR (p)					MA (q)
		C	1	2	3	6	7	8	1
Parameters estimation results		—	$\widehat{\varphi}_1$	$\widehat{\varphi}_2$	$\widehat{\varphi}_3$	$\widehat{\varphi}_6$	$\widehat{\varphi}_7$	$\widehat{\varphi}_8$	$\widehat{\theta}_1$
			0.405	0.197	0.225	0.134	0.110	−0.083	−0.908
Significance test	t-statistic	—	9.086	5.240	6.056	3.502	2.521	−2.079	−32.161
	Prob (t-statistic)	—	<0.001	<0.001	<0.001	<0.001	0.012	0.038	<0.001
R-squared						0.280			
Adjusted R-squared						0.270			

Table 6: SE of the predicted parameters.

	SE.of.quantity	SE.of.speed	SE.of.occupancy
Day 8/10	7.41	4.82	2.81
Day 9/10	7.13	4.46	1.99

$$\begin{cases} \widehat{\text{Quantity}} f_t = \widehat{c} + \sum_{i=1}^{n} \widehat{\varphi}_i * \text{Quantity} f_{t-i} - \sum_{i=2}^{n} \widehat{\theta}_i * e_{t-i} + e_{t-1} if \widehat{\varphi}_i \neq 0 \widehat{\theta}_i \neq 0, \\ \widehat{\text{Speed}} f_t = \widehat{c} + \sum_{i=1}^{n} \widehat{\varphi}_i * \text{Speed} f_{t-i} - \sum_{i=2}^{n} \widehat{\theta}_i * e_{t-i} + e_{t-1} if \widehat{\varphi}_i \neq 0 \widehat{\theta}_i \neq 0, \\ \widehat{\text{Occupancy}} f_t = \widehat{c} + \sum_{i=1}^{n} \widehat{\varphi}_i * \text{Occupancy} f_{t-i} - \sum_{i=2}^{n} \widehat{\theta}_i * e_{t-i} + e_{t-1} if \widehat{\varphi}_i \neq 0 \widehat{\theta}_i \neq 0. \end{cases} \tag{11}$$

Using the parameter prediction model (10), the traffic flow parameters on 8/10/2008 and 9/10/2008 are predicted. The predicted results are measured by the indexSE, as shown in formula (12), and the results are displayed in Table 7. Moreover, the predicted results are shown in Figure 5.

$$SE = \sqrt{\frac{\sum_{i=1}^{n} (y_t - \widehat{y}_t)^2}{n}}. \tag{12}$$

4.3. FISHER Recognition Model Calibration. In Section 3.2, we take the traffic flow parameters from 7/10/2018 as the sample data and use K-means clustering analysis to

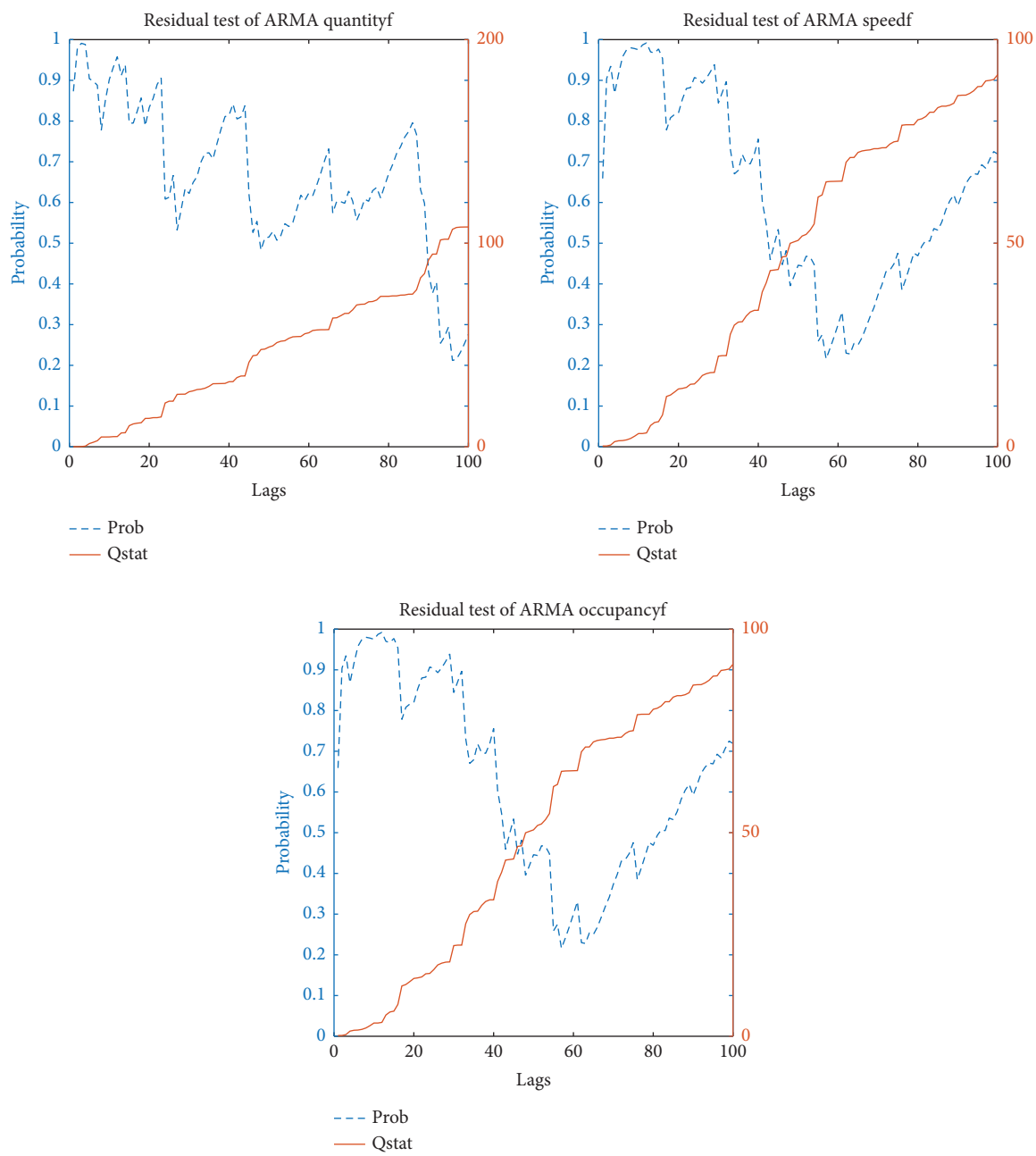

FIGURE 3: Q_{LB} values and corresponding probabilities of a residual sequence.

obtain the classification results of the traffic state. Next, we use this classification result as a training sample and use the Fisher discrimination introduced in Section 2.2 to calibrate the traffic state recognition model.

Case 1. 3 traffic states

As seen from Figure 6, $n_1 = 142, n_2 = 168, n_3 = 170$, and $n = n_1 + n_2 + n_3 = 480$. Let X_1, X_2, and X_3 denote the parameter samples of traffic state 1, traffic state 2, and traffic state 3, respectively: $\overline{x}_1 = \begin{pmatrix} 37.06 \\ 45.22 \\ 6.92 \end{pmatrix}, \overline{x}_2 = \begin{pmatrix} 8.26 \\ 50.68 \\ 1.22 \end{pmatrix}, \overline{x}_3 = \begin{pmatrix} 58.16 \\ 42.19 \\ 12.52 \end{pmatrix}$, and $\overline{x} = \begin{pmatrix} 34.45 \\ 46.06 \\ 6.91 \end{pmatrix}$.

Calculate the sum of squares and cross product sums between groups:

$$B = \sum_{i=1}^{k} n_i (\overline{x}_i - \overline{x})(\overline{x}_i - \overline{x})' = \begin{pmatrix} 211770, -36240, 47650 \\ -36240, 6230, -8110 \\ 47650, -8110, 10790 \end{pmatrix}. \quad (13)$$

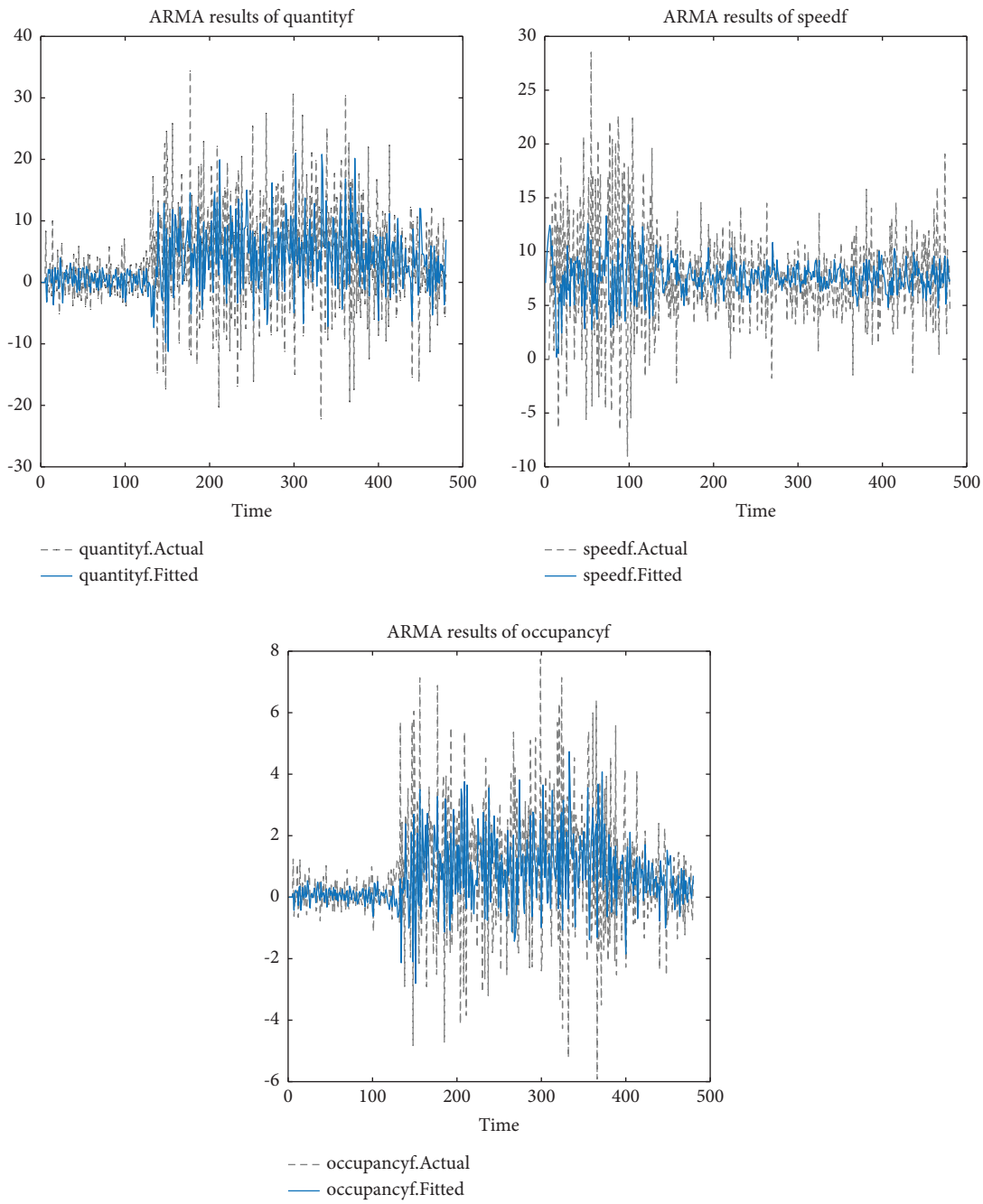

Figure 4: ARMA model fitting results.

Table 7: State prediction results.

State	3 states				5 states					
	State 1	State 2	State 3	Total	State 1	State 2	State 3	State 4	State 5	Total
8/10/2008										
Number of samples	161	167	152	480	101	98	68	81	132	480
Number of correct	113	155	132	400	59	57	42	35	125	318
Accuracy	70.18%	92.81%	86.84%	83.33%	58.41%	58.16%	61.76%	43.21%	94.69%	66.25%
9/10/2008										
Number of samples	142	173	165	480	89	115	59	77	140	480
Number of correct	107	160	139	406	53	74	41	42	130	340
Accuracy	75.35%	92.49%	84.24%	84.58%	59.55%	64.34%	69.49%	54.55%	92.85%	70.83%

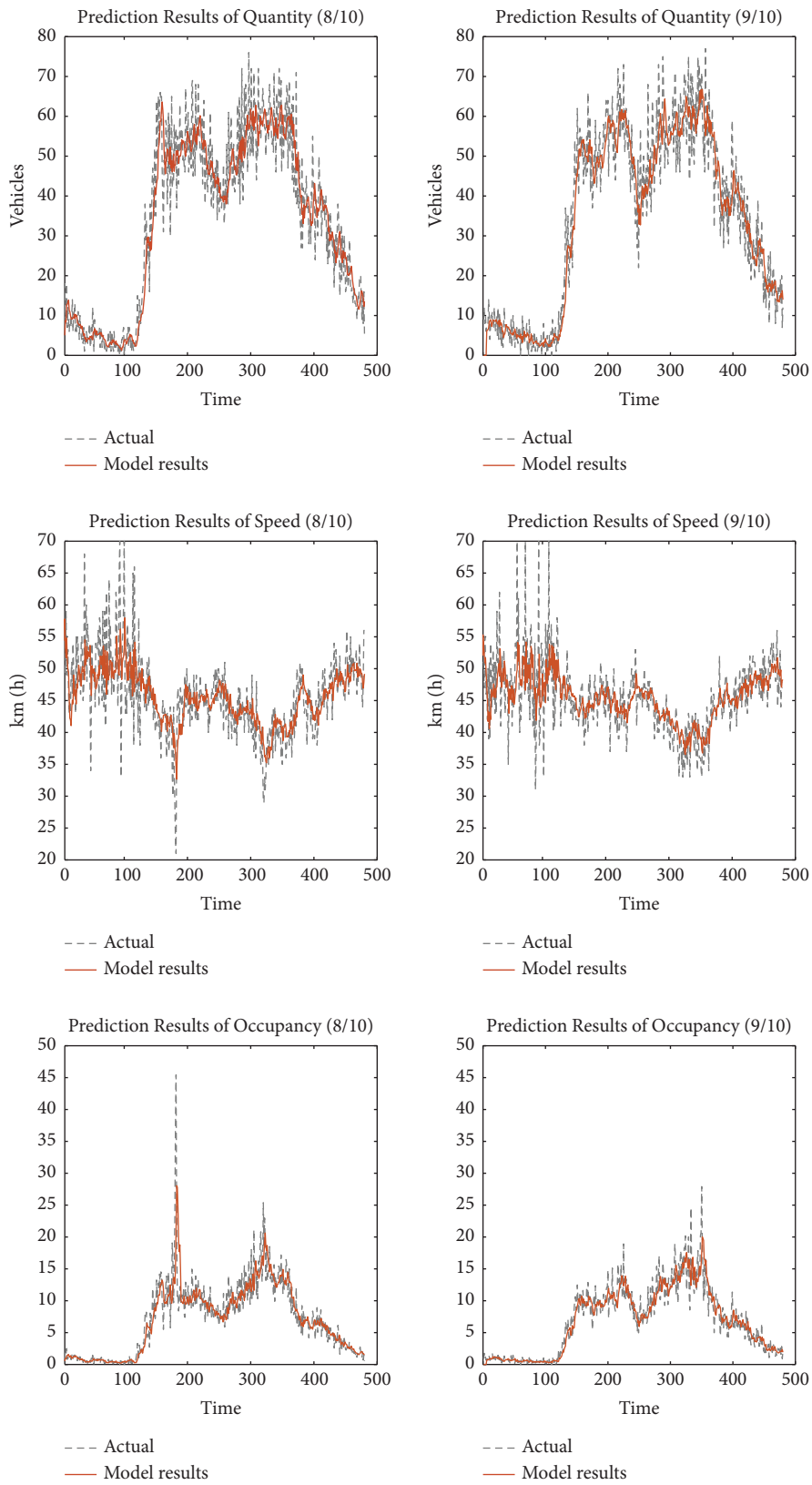

Figure 5: Predicted results of the models.

Calculate the sum of squares and cross product sums within a group:

$$E = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{ij} - \overline{x}_i)(x_{ij} - \overline{x}_i)' = \begin{pmatrix} 20030, -3019, 4270 \\ -3019, 8863, -1167 \\ 4270, -1167, 1362 \end{pmatrix}. \quad (14)$$

Calculate the total nonzero eigenvalues of $E^{-1}B$. Then, we can obtain the eigenvalues $\lambda_1 = 10.64$ *and* $\lambda_2 = 0.17$. The corresponding eigenvectors are $t_1 = \begin{pmatrix} 0.86 \\ -0.02 \\ 0.50 \end{pmatrix}$ *and* $t_2 = \begin{pmatrix} -0.20 \\ 0.09 \\ 0.97 \end{pmatrix}$.

Since $(\lambda_1)/(\lambda_1 + \lambda_2) = 0.98$ is already a very high value, it is sufficient for us to choose the first discriminants. Therefore, the centralized discriminant is as follows:

$$y_1 = 0.86 \times (quantity^* - 34.45) - 0.02 \times (speed^* - 46.06) + 0.50 \times (occupancy^* - 6.91). \quad (15)$$

The group mean of the discriminant is as follows:

$$\begin{cases} \overline{y}_{11} = 2.27, \\ \overline{y}_{21} = -25.46, \\ \overline{y}_{31} = 23.28. \end{cases} \quad (16)$$

The final recognition model can be expressed as an unknown traffic state $\mathbf{x}$ (it is characterized by the three parameters of $quantity^*$, $speed^*$, *and* $occupancy^*$). If $\mathbf{x}$ can satisfy the following conditions, then it belongs to state S_l ($S_1 = state1$, $S_2 = state2$, *and* $S_3 = state3$).

$$\begin{cases} y_1 = 0.86 \times (\text{quantity}^* - 34.45) - 0.02 \times (\text{speed}^* - 46.06) + 0.50 \times (\text{occupancy}^* - 6.91), \\ \mathbf{x} \in S_l,\ if\ (y_1 - \overline{y}_{l1})^2 = \min_{1 \le i \le 3} (y_1 - \overline{y}_{i1})^2. \end{cases} \quad (17)$$

Case 2. 5 traffic states

In the same way, the recognition model in five states can be obtained, such as formulas (18) and (19).

The group mean of the discriminant is as follows:

$$\begin{cases} \overline{y}_{11} = -3.02, \\ \overline{y}_{21} = -17.79, \\ \overline{y}_{31} = 11.70, \\ \overline{y}_{41} = -30.48, \\ \overline{y}_{51} = 29.17. \end{cases} \quad (18)$$

The final recognition model can be expressed as an unknown traffic state $\mathbf{x}$, If $\mathbf{x}$ can satisfy the following conditions, then it belongs to state S_l ($S_1 = state1$, $S_2 = state2$, $S_3 = state3$, $S_4 = state4$, *and* $S_5 = state5$).

$$\begin{cases} y_1 = -0.96 \times (quantity^* - 34.45) + 0.09 \times (speed^* - 46.06) - 0.26 \times (occupancy^* - 6.92), \\ \mathbf{x} \in S_l,\ if\ (y_1 - \overline{y}_{l1})^2 = \min_{1 \le i \le 5} (y_1 - \overline{y}_{i1})^2, \end{cases} \quad (19)$$

4.4. Discrete Traffic States Prediction Results. Using the traffic flow parameter prediction data obtained in Section 4.2 (8/10/2008 and 9/10/2008) and the identification models of (17) and (19), the corresponding traffic conditions can be predicted. The prediction results are shown in Table 3, Figures 6 and 7. Figure 6 shows the prediction results for three classification states, and Figure 7 shows the prediction results for five classification states. We also compared the state prediction results with the original state. The results show that in the three states, the prediction accuracy for 8/10/2008 can reach 83.33% and the prediction accuracy for 9/10/2008 can reach 84.58%. Under the three states situation, the highest prediction accuracy is State 2 (the smoothest state), and the accuracies reached 92.81% and 92.49% for the prediction results on 8/10/2008 and 9/10/2008, respectively. The prediction results for the same dates in the five state situations are also confirmed. The highest prediction accuracy is state 5 (the smoothest state), and the accuracies are 94.69% and 92.85%, respectively. We speculate that this result may be caused by the most obvious state being different from other states. In the five states situation, the prediction accuracy for 8/10/2008 can reach 66.25% and the prediction accuracy for 9/10/2008 can reach 70.83%, which are significantly lower than the prediction accuracies of the three-state situations. This is because more states result in less obvious the differences

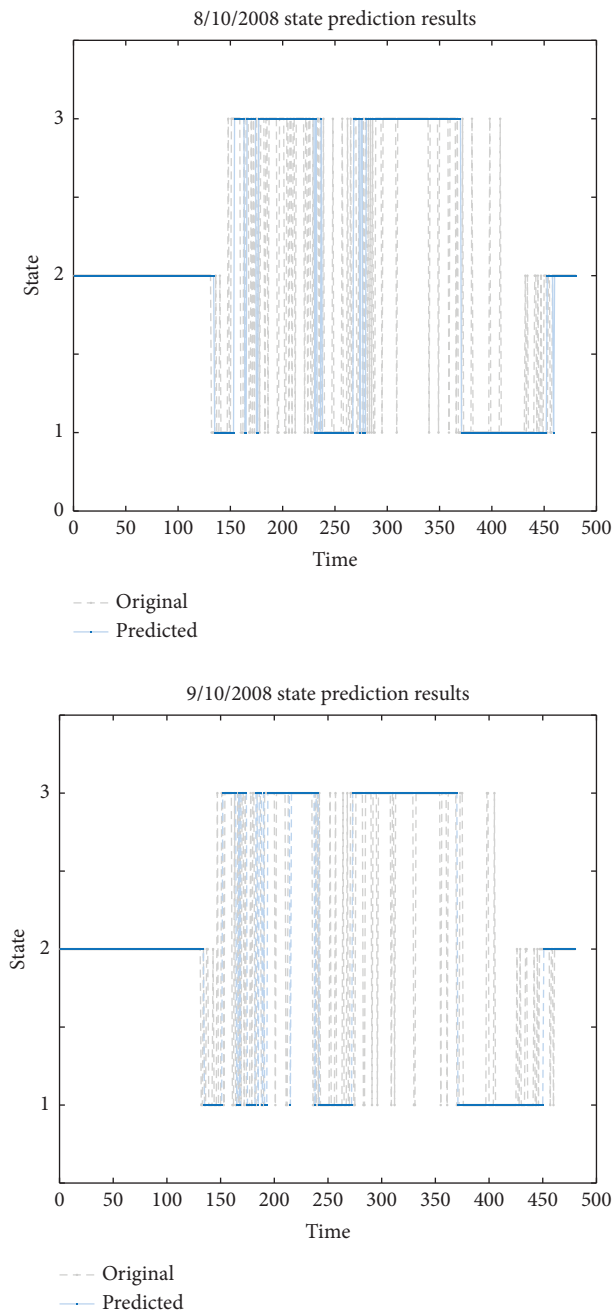

Figure 6: Three traffic states prediction results.

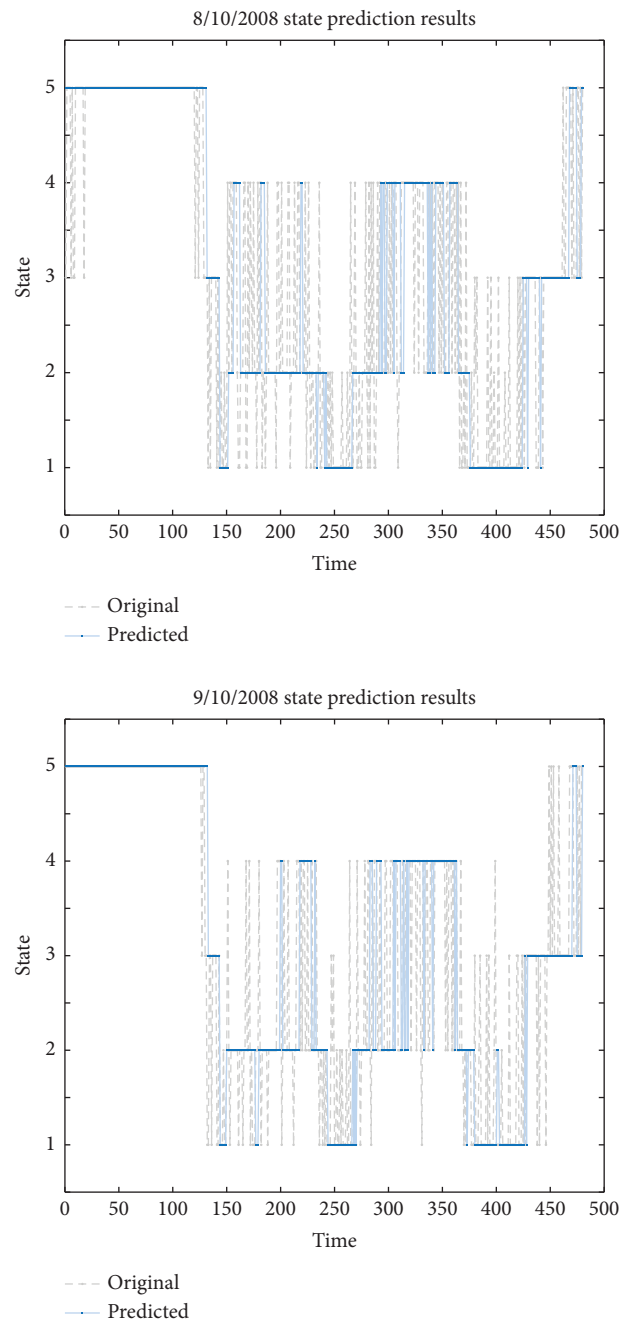

Figure 7: Five traffic states prediction results.

between these states, leading to a high misjudgment of each state.

4.5. Result Analyses. In order to evaluate the effectiveness of the prediction model proposed in this paper, we compare the prediction results with those of previous similar prediction methods. These methods mainly include parameter prediction methods, such as ARIMA [28], ARIMA-GARCH [35], ARIMAX [29], and hybrid methods such as ARIMA-ANN [9]. The principles and details of these models can be referred to in the relevant literature and will not be described here. Here, we first compare the parameter prediction results with GFD-ARMA and then compare the state prediction results using the GFD-ARMA-FISHER analytical framework. The parameters of ARIMA (p, d, q) is set by ARIMA (3, 1, 2) (for Quantity)\ARIMA (4, 1, 1) (for Speed)\ARIMA (5, 1, 1) (for Occupancy). The parameters of ARIMA-GARCH is set by ARIMA (3, 1, 2)-GARCH (1, 1) (for Quantity)\ARIMA (4, 1, 1)-GARCH (1, 1) (for Speed)\ARIMA (5, 1, 1)-GARCH (1, 1) (for Occupancy). The input variable of the ARIMAX is Headway, the parameters is set by ARIMAX (3, 1, 1). Specific details of ARIMA-ANN prediction model are shown in the literature [9], and the ANN part adopts the BP neural network with double hidden layers (the network structure is $5 \times 6 \times 4 \times 1$), 5 inputs and 1

Table 8: Comparison results of the predicted parameters.

Data		Methods				
		GFD-ARMA	ARIMA	ARIMA-GARCH	ARIMAX	ARIMA-ANN
Day 8/10	SE of quantity	7.41	13.18	9.15	14.35	**6.71**
	SE of speed	4.82	12.79	9.82	15.89	5.85
	SE of occupancy	2.81	12.65	7.63	12.64	5.98
Day 9/10	SE of quantity	7.13	13.98	9.17	15.31	**6.91**
	SE of speed	4.46	12.64	8.69	15.22	5.86
	SE of occupancy	1.99	13.13	7.48	15.5	5.63

Table 9: Comparison results of the predicted state.

Data			State prediction results based on FISHER				
			GFD-ARMA	ARIMA	ARIMA-GARCH	ARIMAX	ARIMA-ANN
3 states	Day 8/10	Number of samples	480	480	480	480	480
		Number of correct	400	369	373	360	389
		Accuracy	83.33%	76.87%	77.7%	75%	81%
	Day 9/10	Number of samples	480	480	480	480	480
		Number of correct	406	362	369	353	385
		Accuracy	84.58%	75.41%	76.88%	73.54%	80.2%
5 states	Day 8/10	Number of samples	480	480	480	480	480
		Number of correct	318	281	282	266	302
		Accuracy	66.25%	58.54%	58.75%	55.42%	62.91%
	Day 9/10	Number of samples	480	480	480	480	480
		Number of correct	340	276	279	261	300
		Accuracy	70.83%	57.5%	58.13%	54.38%	62.5%

output are designed. The number of double hidden layer neurons is 6 and 4, respectively. The results of parameter prediction and state identification are shown in Tables 8 and 9, respectively.

As can be seen from Tables 8 and 9: (1) For the quantity parameter, the GFD-ARMA model performs better than ARIMA-GARCH and ARIMAX, but worse than ARIMA-ANN model. (2) For the speed and occupancy, the GFD-ARMA model performs better than all of other models. (3) Although ARIMA-ANN has better prediction results for individual parameters (Quantity) than GFD-ARMA, GFD-ARMA -Fisher shows best in predicting traffic state. This also shows the advantages of GFD-ARMA-Fisher prediction framework in traffic state prediction.

5. Conclusions

The accurate prediction of traffic states is of great significance for traffic information managers to monitor road traffic conditions, provide traffic guidance information, and assist traffic engineering departments in planning and improving traffic control systems. Based on the analysis of previous studies, this paper proposes decomposing the traffic states prediction into two parts: the prediction of traffic parameters and the identification of traffic states. Based on this idea, this paper proposes a discrete traffic state prediction model based on the GFD-ARMA-FISHER analytical framework. The prediction results show that in the three states situation, the prediction accuracy for 8/10/2008 and 9/10/2008 can reach 83.33% and 84.58%, respectively. In the five states situation, the prediction accuracies for 8/10/2008 and 9/10/2008 can reach 66.25% and 70.83%, respectively, which are significantly lower than the prediction accuracies in the three states situation. We speculate that this is because having more states results in less obvious differences between these states, leading to a high misjudgment of each state. We also compare the prediction method proposed in this paper with the existing research results, and the results show that the method proposed in this paper has obvious advantages in the prediction of the final traffic state.

Data Availability

The data of this article can be obtained by contacting the author's e-mail (danis_cx@126.com).

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This paper is supported by the Innovation Project of University of Shanghai for Science and Technology (XJ2022151).

References

[1] M. Rodriguez-Vega, C. Canudas-De-Wit, and H. Fourati, "Urban network traffic state estimation using a data-based approach," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 278–283, 2021.

[2] N. Alemazkoor and H. Meidani, "A data-driven multi-fidelity approach for traffic state estimation using data from multiple sources," *IEEE Access*, vol. 9, p. 1, 2021.

[3] K. Zhang, X. Feng, N. Jia, L. Zhao, Z. He, and T. Gan, "TSR-GAN: generative adversarial networks for traffic state reconstruction with time space diagrams," *Physica A: Statistical Mechanics and Its Applications*, vol. 591, Article ID 126788, 2022.

[4] W. Jiang, Y. Xiao, Y. Liu, Q. Liu, Z. Li, and B. Grcn, "A spatio-temporal traffic flow prediction model based on graph neural network," *Journal of Advanced Transportation*, vol. 2022, Article ID 5221362, 2022.

[5] Z. Zheng and D. Su, "Short-term traffic volume forecasting: a k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 143–157, 2014.

[6] H. Qian, Y. Z. Chen, Z. L. Li, and Y. H. Yang, "Comparison of kernel approach and knn approach in short-term forecasting of traffic flow," *Computer and Communications*, vol. 6, pp. 18–34, 2008.

[7] L. Zhang, Y. Feng, L. Li, and X. Bai, "Highway traffic flow prediction based on optimized KNN of spark," in *Proceedings of the 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, Chengdu, China, May 2021.

[8] H. Zeng, Z. Peng, X. H. Huang, Y. Yang, and R. Hu, "Deep spatio-temporal neural network based on interactive attention for traffic flow prediction," *Applied Intelligence*, vol. 52, pp. 1–12, 2022.

[9] M. C. Tan, L. B. Feng, and J. M. Xu, "Traffic flow prediction based on hybrid arima and ann model," *China Journal of Highway and Transport*, vol. 20, 2007.

[10] A. Cheng, X. Jiang, Y. Li, C. Zhang, and H. Zhu, "Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method," *Physica A: Statistical Mechanics and Its Applications*, vol. 466, pp. 422–434, 2017.

[11] M. W. Li, W. C. Hong, and H. G. Kang, "Urban traffic flow forecasting using Gauss–SVR with cat mapping, cloud model and PSO hybrid algorithm," *Neurocomputing*, vol. 99, pp. 230–240, 2013.

[12] J. Liu, X. Wang, Y. Li, X. Kang, and L. Gao, "Method of evaluating and predicting traffic state of highway network based on deep learning," *Journal of Advanced Transportation*, vol. 2021, Article ID 8878494, 9 pages, 2021.

[13] Z. Mingheng, Z. Yaobao, H. Ganglong, and C. Gang, "Accurate multisteps traffic flow prediction based on SVM," *Mathematical Problems in Engineering*, vol. 2013, Article ID 418303, 8 pages, 2013.

[14] N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[15] Q. Liu, Y. Cai, H. Jiang, J. Lu, and L. Chen, "Traffic state prediction using ISOMAP manifold learning," *Physica A: Statistical Mechanics and Its Applications*, vol. 506, pp. 532–541, 2018.

[16] A. Bezuglov and G. Comert, "Short-term freeway traffic parameter prediction: application of grey system theory models," *Expert Systems with Applications*, vol. 62, pp. 284–292, 2016.

[17] C. Chen, X. Liu, T. Qiu, and A. K. Sangaiah, "A short-term traffic prediction model in the vehicular cyber–physical systems," *Future Generation Computer Systems*, vol. 105, 2017.

[18] S. Alkheder and A. Alomair, "Urban traffic prediction using metrological data with fuzzy logic, long short-term memory (LSTM), and decision trees (DTs)," *Natural Hazards*, vol. 111, no. 2, pp. 1685–1719, 2022.

[19] Z. Yang, Q. Bing, C. Lin, N. Yang, D. Mei, and J. University, "Research on short-term traffic flow prediction method based on similarity search of time series," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–8, 2014.

[20] J. Wang, W. Deng, and Y. Guo, "New Bayesian combination method for short-term traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 79–94, 2014.

[21] L. Mannini, S. Carrese, E. Cipriani, and U. Crisalli, "On the short-term prediction of traffic state: an application on urban freeways in ROME," *Transportation Research Procedia*, vol. 10, pp. 176–185, 2015.

[22] Y. Zhang, Y. Zhang, and A. Haghani, "A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 65–78, 2014.

[23] Q. Shang, C. Lin, Z. Yang, Q. Bing, and X. Zhou, "A hybrid short-term traffic flow prediction model based on singular spectrum analysis and kernel extreme learning machine," *PLoS One*, vol. 11, no. 8, Article ID e0161259, 2016.

[24] T. T. Tchrakian, B. Basu, and M. O'Mahony, "Real-time traffic flow forecasting using spectral analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 519–526, 2012.

[25] Z. Cao and L. Lei, "The improvement of the forecasting model of short-term traffic flow based on wavelet and arma," in *Proceedings of the 2010 8th International Conference on Supply Chain Management and Information*, Hong Kong, China, October 2010.

[26] D. R. Huang, J. Song, D. C. Wang, J. Q. Cao, and W. Li, "Forecasting model of traffic flow based on ARMA and wavelet transform," *Computer Engineering and Applications*, vol. 42, pp. 191–224, 2006.

[27] G. Yang, R. Xv, M. Qin, K. Zheng, and B. Zhang, "Short-term traffic volume forecasting based on ARMA and kalman filter," *Journal of Zhengzhou University Engineering Science*, vol. 38, pp. 36–40, 2017.

[28] Y. F. Yao and C. Feng, "Short-time traffic flow prediction based on arima," *Technology & Economy in Areas of Communications*, vol. 35, 2006.

[29] M. Yang, J. Xie, P. Mao, W. Chao, and Z. Ye, "Application of the ARIMAX model on forecasting freeway traffic flow," in *Proceedings of the Conference: 17th COTA International Conference of Transportation Professionals*, Shanghai, China, March 2017.

[30] Y. Zhou, Q. Nie, J. Xia, and C. An, "Traffic Volatility Forecasting for Urban Arterials: Family GARCH Approach," *Transportation Research Board Annual Meeting*, Washington, DC, USA, 2015.

[31] Q. P. Wang, S. Qin, and H. G. Huang, "Study on GARCH Effect in Traffic Flow Data and Prediction Model," *Computer Simulation*, vol. 33, 2016.

[32] Y. Rajabzadeh, A. H. Rezaie, and H. Amindavar, "Short-term traffic flow prediction using time-varying Vasicek model," *Transportation Research Part C: Emerging Technologies*, vol. 74, pp. 168–181, 2017.

[33] Q. Cui and J. Xia, "Time-varying confidence interval forecasting of travel time for urban arterials using ARIMA-GARCH model," *Journal of Southeast University*, vol. 30, pp. 358–362, 2014.

[34] C. Ding, J. Duan, Y. Zhang, X. Wu, and G. Yu, "Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic

volatility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1054–1064, 2018.

[35] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model," *IEEE Intelligent Vehicles Symposium*, vol. IV, pp. 607–612, 2011.

[36] L. Hu, T. Huang, X. Shi et al., "Predicting functions of proteins in mouse based on weighted protein-protein interaction network and protein hybrid properties," *PLoS One*, vol. 6, no. 1, Article ID e14556, 2011.

[37] I. Laña, J. L. Lobo, E. Capecci, J. Del Ser, and N. Kasabov, "Adaptive long-term traffic state estimation with evolving spiking neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 126–144, 2019.

[38] Y. Hara, J. Suzuki, and M. Kuwahara, "Network-wide traffic state estimation using a mixture Gaussian graphical model and graphical lasso," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 622–638, 2018.

[39] A. Takenouchi, K. Kawai, and M. Kuwahara, "Traffic state estimation and its sensitivity utilizing measurements from the opposite lane," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 95–109, 2019.

[40] A. Bhaskar, T. Tsubota, L. M. Kieu, and E. Chung, "Urban traffic state estimation: fusing point and zone based data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 120–142, 2014.

[41] D. Gillings, D. Makuc, and E. Siegel, "Analysis of interrupted time series mortality trends: an example to evaluate regionalized perinatal care," *American Journal of Public Health*, vol. 71, pp. 38–46, 1981.

[42] S. J. Koopman, M. Ooms, and M. A. Carnero, "Periodic seasonal reg-arfima-garch models for daily electricity spot prices," *Publications of the American Statistical Association*, vol. 102, pp. 16–27, 2007.

[43] V. A. Reisen, "ESTIMATION OF THE FRACTIONAL DIFFERENCE PARAMETER IN THE ARIMA(p, d, q) MODEL USING THE SMOOTHED PERIODOGRAM," *Journal of Time Series Analysis*, vol. 15, no. 3, pp. 335–350, 1994.