

Traffic Index Prediction and Classification Considering Characteristics of Time Series Based on Autoregressive Integrated Moving Average Convolutional Neural Network Model

Jian Lu,^{1,2} Xuedong Zhang,^{1,2*} Zhijie Xu,^{1**} Jianqin Zhang,^{1,2}
Jingjing Wang,³ Lizeng Mao,³ Lipeng Jia,^{1,2} and Zhuohang Li^{1,2}

¹Beijing University of Civil Engineering and Architecture,
15 Yongyuan Rd., Daxing District, Beijing 102616, China

²Key Laboratory for Urban Spatial Informatics of Ministry of Natural Resources,
15 Yongyuan Rd., Daxing District, Beijing 102616, China

³Beijing Municipal Transportation Coordination Center, Beijing 100073, China

(Received August 11, 2020; accepted October 20, 2020)

Keywords: traffic index, time series, ARIMA-CNN model, prediction and early warning

We propose an autoregressive integrated moving average convolutional neural network (ARIMA-CNN) to address the problems of the large amount of computation required for typical networks, the low effectiveness of traditional machine learning in traffic index prediction, and the weak recognition ability of traditional methods based on distance and features to improve the traffic index in traffic scheduling. The ARIMA-CNN can accurately predict the traffic index and distinguish its model categories. The model includes two steps: traffic index prediction and prediction index classification. The first step uses the augmented Dickey–Fuller (ADF) test to determine the type of traffic index series and then converts a nonstationary series to a stationary series by the difference operation. The ARIMA is fitted with the Bayesian information criterion (BIC) matrix, and the traffic index is predicted by the ARIMA. The second step obtains the best CNN model based on the traffic index feature information extracted from the training time series, integrates the feature information into a one-dimensional feature vector, determines the feature vector pattern category according to the Softmax classifier, and decides the category of the predicted traffic index. We used the traffic index data of Beijing for three consecutive years (from 2016 to 2018) as an example. The time series of the traffic index in the experimental data was accurately predicted, the prediction results were consistent with the variation characteristics of the real series data, and the prediction mode of recognition was the Monday mode. Our results were consistent with the actual situation and prove the validity of the model. On the basis of the identification results and the corresponding threshold curve of this category, we were able to find abnormal points, supervise the traffic status, and obtain early warnings about possible abnormal traffic patterns. This research has important practical significance for helping traffic management departments to make traffic control decisions in advance.

*Corresponding author: e-mail: zhangxuedong@bucea.edu.cn

**Corresponding author: e-mail: xuzhijie@bucea.edu.cn

<https://doi.org/10.18494/SAM.2020.3058>

1. Introduction

Transportation is the lifeblood of a city. The rapid development of the urban economy has resulted in an increase in traffic congestion, particularly in economically developed areas. To cope with complex and changing traffic conditions and relieve the traffic pressure in the city, traffic management departments have issued traffic laws and regulations to restrict driving. Research institutions use the Internet of Things and other technologies to assist traffic management departments in monitoring road congestion. Although various industries have made some progress toward reducing the incidence of traffic congestion, the negative impact of traffic congestion on a city cannot be entirely eliminated because of the complex environment, emergencies, human behavior, and other factors. The traffic index is an important index for studying the status of urban traffic. It reflects the quantitative results of urban congestion and has some rules concerning time. The characteristics of residents' travel can be obtained from the historical traffic index through time series pattern recognition, and residents' travel can be classified, providing basic data for the prediction of traffic patterns. Therefore, this research has important value for urban traffic congestion relief.

At present, the prediction methods of the traffic flow index mainly include the recursive algorithm of the Kalman filter, grey system, support vector machine, and deep learning. Although the iterative estimation model based on the recursive algorithm of the Kalman filter has been widely used in passenger flow prediction, it requires numerous matrix and vector operations, resulting in a low efficiency.⁽¹⁻³⁾ The grey system model can make predictions by identifying the different degrees of the development trend among the system factors; it has a positive effect on the short-term prediction of data but a negative effect on the long-term prediction of data.^(4,5) The support vector machine needs to map the input layer to a high-dimensional space and solve the separation hyperplane, requiring a significant amount of computation.⁽⁶⁻⁸⁾ The long short-term memory model alleviates the problem of the gradient disappearance in the cyclic neural network model, but it needs a linear layer to run in each series of time steps. The layer needs a large number of storage bandwidth calculations and a significant amount of training.⁽⁹⁻¹¹⁾ The methods of traffic flow pattern recognition mainly include distance-based pattern recognition and feature-based pattern recognition. Distance-based pattern recognition generally uses the Euclidean distance to measure the similarities of the traffic flow, and a typical example is the K-nearest neighbor algorithm.^(12,13) A feature-based pattern recognition algorithm generally looks for different subsegments to distinguish traffic flow categories. For example, the shapelet algorithm looks for the most representative continuous subseries in the data.⁽¹⁴⁾ Although the two kinds of pattern recognition methods can achieve good classification results under specific conditions, they are affected by many traffic factors, and there are some distortions and deformations in the traffic index time series data itself. Therefore, the traditional distance-based and feature-based recognition methods still have some shortcomings in regard to the pattern recognition ability of the traffic index time series data.

We propose a method based on the autoregressive integrated moving average (ARIMA) and convolutional neural network (CNN) models (ARIMA-CNN) to realize the prediction and

pattern classification of traffic index time series data. The model first uses the ARIMA to predict the traffic index, and then classifies the traffic index prediction results to determine the future traffic mode. The prediction curve is compared with the corresponding mode threshold curve to find the abnormal points and support predictions and early warnings. This model can focus on the abnormal traffic conditions at a certain moment, and it can provide data-driven support for traffic management departments to help them give early warnings.

2. Autoregressive Integrated Moving Average

The ARIMA algorithm is a model established by the regression of the lag value of the dependent variable and the present value and the lag value of the random error term in the process of converting a nonstationary time series to a stationary time series.⁽¹⁵⁾ The basic idea of the algorithm is to treat the data formed by the prediction object over time as a random series, use a mathematical model to describe the series, and then use the model to predict the future value from the past value, which is composed of an autoregressive process, averaging process, and difference process. ARIMA uses the different stationarity of time series data and part of the regression analysis. ARIMA includes the autoregressive (AR) model, moving average (MA) model, and autoregressive moving average (ARMA) model. ARIMA's prediction process for time series data is shown in Fig. 1.

2.1 Stability test

Before predicting the time series data, data preprocessing is required, including a randomness test and a stationarity test. According to the test results, the data are divided into three types: a purely random series, a stationary nonrandom series, and a nonstationary nonrandom series. A purely random series is also called a white noise series, which means that there is no relationship between the series items. The series is completely disordered and randomly distributed, and it has no research value. The mean and variance of a stationary nonrandom series are constants. Generally, a linear model is used to fit the data, and then the series rules can be extracted. The mean and variance of nonstationary and nonrandom series are uncertain, and they need to be transformed into a stationary series through the difference operations. The mean and variance of the stationary series data will not change significantly within the time period, and the fitting curve of the time series maintains the existing form for a

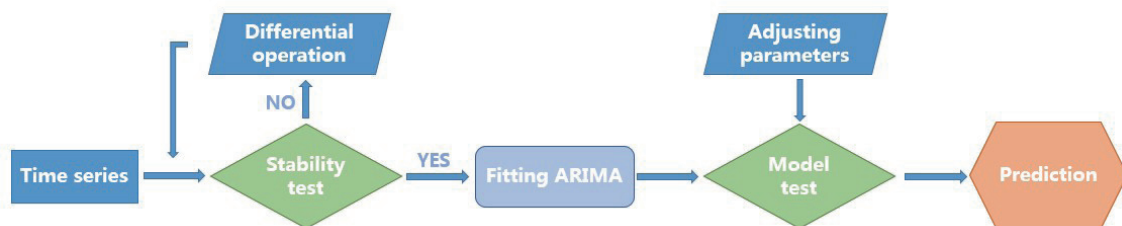


Fig. 1. (Color online) Flowchart of ARIMA.

short time. Stationary series include those that are strict stationary and generalized stationary. Strict stationary means that the data distribution will not change over time, and generalized stationary means that the expectation and correlation coefficients of the series data will not change. Strict stationary is often too absolute; the vast majority of real-world instances are generalized stationary.

The pure randomness test is conducted by constructing test statistics and calculating the p value corresponding to the test statistics. If the p value is greater than the significance level α , it indicates a purely random series. Because of the mean and variance, the time series has a limited range of fluctuations at a certain time for nonrandom series. If the periodic autocovariance is equal to the autocorrelation coefficient, the time series is considered stationary. Generally, a timing diagram and the augmented Dickey–Fuller (ADF) test are used to judge the nature of a time series.

2.2 Difference operation

The nonstationary time series needs to be transformed into a stationary time series by the difference operation.⁽¹⁶⁾ We set $\{x_t | t \in T\}$ to be a set of time series data. B is the backward operator, i.e., $Bx_t = x_{t-1}$. $\nabla = 1 - B$ is called the difference operator.

$$\nabla x_t = (1 - B)x_t = x_t - x_{t-1} \quad (1)$$

Generally,

$$\nabla^d x_t = (1 - B)^d x_t = \sum_{i=0}^d (-1)^i C_d^i x_{t-i}. \quad (2)$$

From Eqs. (1) and (2), the stationary time series can be obtained by the difference operation of the nonstationary time series data.

2.3 Autoregressive operation

The autoregressive operation is used to describe the relationship between the current value and the historical value, and this approach uses its own historical data to predict itself.⁽¹⁷⁾ The stability of the data is determined by autoregression, and the data used must be autorelevant. If the autocorrelation coefficient is less than 0.5, it is not appropriate to use autoregressive calculation. The autoregressive operation is only applicable to the prediction of phenomena related to its earlier stages, and its expression is

$$y_t = \mu + \sum_{i=1}^p \gamma_i y_{t-i} + \varepsilon_t, \quad (3)$$

where y_t is the current value, μ is the constant term, p is the order, γ_i is the autocorrelation coefficient, and ε_t is the error value.

2.4 Moving average process

The autoregressive calculation causes the accumulation of error terms. Therefore, the moving average is required to eliminate random fluctuations in predictions by Eq. (4):

$$y_t = \mu + \varepsilon_t + \sum_{i=1}^p \theta_i \varepsilon_{t-i}, \quad (4)$$

where y_t is the current value, μ is the constant term, p is the order, ε_t is the error value, and θ_i is the correlation coefficient in the moving average process.

3. CNNs

To mine the deep-level laws and characteristics of time series data (due to the difficulty of extracting the characteristics of the time series data by ordinary clustering algorithms), we used a CNN to realize the pattern recognition of the traffic index. A CNN is a neural network that implements image processing through backpropagation.⁽¹⁸⁾ As shown in Fig. 2, it includes an input layer, convolutional layer, pooling layer, fully connected layer, and output layer.

3.1 Input layer and output layer

In CNNs, the input layer consists of multidimensional data. The input layer of a one-dimensional CNN is generally a one-dimensional or two-dimensional array; the input layer of a two-dimensional CNN is a two-dimensional or three-dimensional array; and the input layer of a three-dimensional CNN is a four-dimensional array. CNNs are widely used in computer image processing. Therefore, the input layer in many studies is a two-dimensional pixel or RGB image. The output layer can output the size and coordinates of the object and its classification.

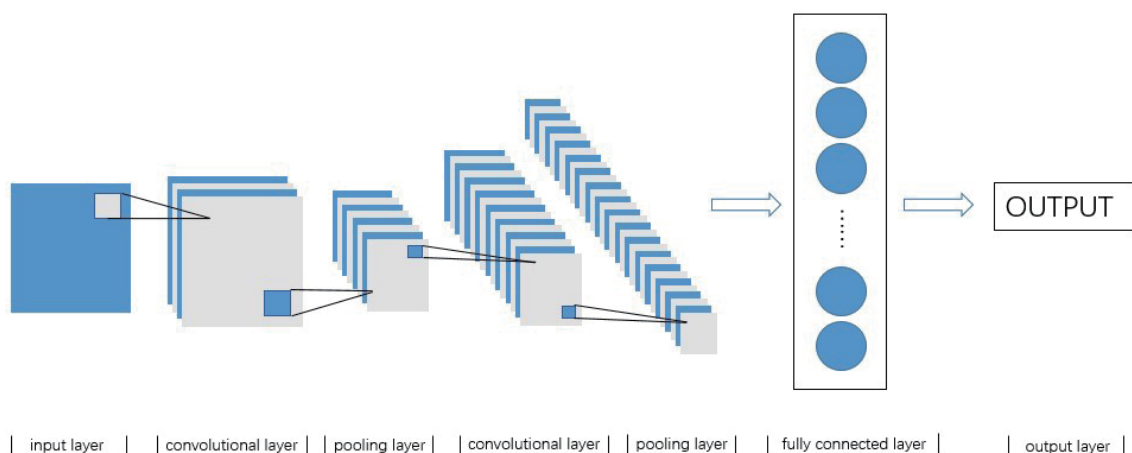


Fig. 2. (Color online) Schematic diagram of the CNN structure.

3.2 Convolutional layer and pooling layer

The function of the convolutional layer is to traverse the input layer through the internal convolution kernel to extract features.⁽¹⁹⁾ The algorithm and size of the convolution kernel are determined according to the size of the input layer, and the convolution kernel moves a fixed unit of length on the input layer each time, as shown in Fig. 3. Equation (5) is the expression for the convolution operation between the input layer and the convolution kernel.

$$a_j^l = f\left(\sum_{i \in M_i} a_i^{l-1} K_j^l + b_j^l\right) \tag{5}$$

Here, a_j^l is the output result after the convolution operation, K_j^l is the j th convolution of the l th layer, b_j^l is the offset parameter, and $f(x)$ is the activation function.

To solve a linear inseparable problem, the result after the convolution operation must use the activation function for the nonlinear transformation. Common activation functions include the sigmoid function [Eq. (6)], tanh function [Eq. (7)], and ReLU function [Eq. (8)].⁽²⁰⁾ The sigmoid function and tanh function include a large number of calculations and can lead to the disappearance of the gradient or gradient explosion, resulting in information loss, which is not conducive to neural network training. The ReLU function can effectively reduce the gradient disappearance and gradient explosion, thereby optimizing the calculation process, reducing parameter dependences, and reducing the probability of overfitting. Therefore, we used the ReLU function to realize the nonlinear transformation after the convolution operation.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{7}$$

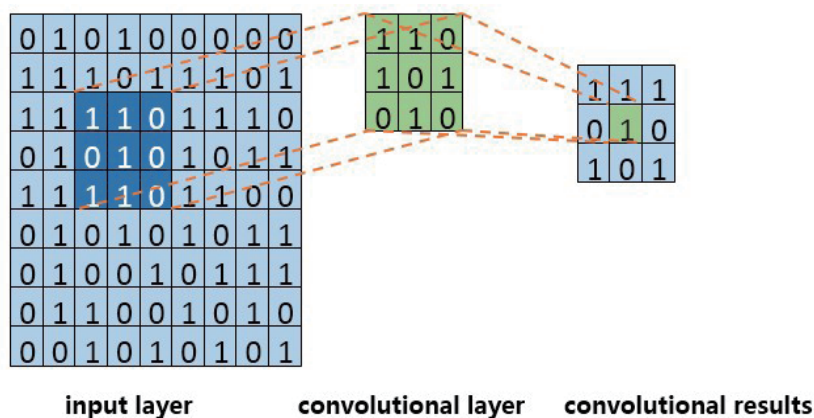


Fig. 3. (Color online) Convolution operation.

$$ReLU(x) = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases} \quad (8)$$

The function of the pooling layer is to further compress the feature layer after convolution processing.⁽²¹⁾ Common pooling layers are divided into mean pooling and maximum pooling. Mean pooling involves taking the mean value of the feature points in the neighborhood as the feature description of the neighborhood; maximum pooling involves taking the maximum value of the feature points in the neighborhood as the feature description of the neighborhood. For data features, mean pooling affects the size of the feature values through the mean, resulting in a loss of feature information. Maximum pooling not only retains most features of the values in the data, but also reduces the error of the parameter adjustment for the mean shift to retain more feature information of the input layer. The pooling layer can reduce dimensions, remove redundant information, compress features, simplify network complexity, and reduce computation and memory consumption. Therefore, selecting an appropriate pooling layer can highlight the advantages of a CNN in image processing according to the actual needs of different experiments.

3.3 Fully connected layer and Softmax function

After processing by the convolutional layer and the pooling layer, the feature information of the input layer is extracted, and the feature information is classified after entering the fully connected layer. The fully connected layer integrates the feature layers, so that the local features are integrated in high dimensions, and a feature vector integrating all the input layer feature information is output.^(22,23) For the integrated vectors, we used the Softmax function for classification,⁽²⁴⁾ calculate the probability of each category using Eq. (9), and determine the corresponding category with the largest probability value as the classification result according to the magnitude of the probability:

$$S_j = \frac{e^{a_j}}{\sum_{k=1}^T e^{a_k}}, \quad (9)$$

where a_j represents the j th value of the vector in the fully connected layer, a_k represents each value in the fully connected layer, and T represents the number of preset classification categories.

4. Time Series Data

Time series data refer to a series of random variables that change with time, which differ from ordinary data in that time affects the data. Depending on the time of the observation, the time interval in the time series can be the year, month, date, or any other time interval. Time series data are very common in finance (such as the stock trading volume) and transportation (traffic passenger flow). Assuming a set of random variables $X = \{X_1, X_2, \dots, X_n\}$ and the definition of time $T = \{t_1, t_2, \dots, t_n\}$, we define $X_t = \{X_1, X_2, \dots, X_t \mid t \in T\}$ as the time series within time T .

The traffic index is a quantitative indicator to measure urban traffic that shows the state of urban traffic at the corresponding time. By monitoring the urban traffic status, the traffic management department translates objective measures, such as the real-time traffic volume, speed, and congestion status, into 0–10 traffic status evaluation indicators. The time series data used in this article are from the Beijing traffic index for November 2016, November 2017, and November 2018 (excluding the data of November 30, 2018). The time interval frequency was 15 min, and the traffic index was recorded between 05:00 and 23:00, a total of 73 values per day. Larger values indicate more traffic congestion and poorer traffic conditions. Table 1 shows the partial traffic index of each time period in Beijing. The time series data of the traffic index show significant differences at different times and dates. As shown in Table 2, the traffic index model was divided into the Monday mode, midweek mode, Friday mode, Saturday mode, and Sunday mode by factors such as the peak value, valley value, location of the second peak, value of the second peak, and duration of the second peak.

5. Case Analysis

5.1 Index prediction

The ARIMA model is good at short-term time series prediction, and the experimental data in this article were 1×73 -dimensional time series data. The data time span was large. Therefore, we divided the day data into seven time periods: 05:00–07:15 (10 moments), 07:30–09:45 (10

Table 1
Beijing traffic index chart.

Date	05:00	...	07:45	08:00	08:15	...	17:30	17:45	18:00	...	23:00
2016.11.1	1.0	...	5.5	6.9	7.4	...	6.7	7.6	7.7	...	1.3
2016.11.2	0.8	...	5.5	6.7	7.5	...	7.3	7.8	8.0	...	1.5
2016.11.3	0.8	...	5.3	6.3	7.0	...	6.2	7.7	7.9	...	1.5
...
...
2018.11.27	1.3	...	6.0	6.5	6.5	...	4.5	6.0	6.8	...	4.5
2018.11.28	1.1	...	5.3	6.5	6.9	...	5.9	7.1	7.9	...	1.5
2018.11.29	1.4	...	4.5	5.8	6.3	...	4.1	6.1	6.8	...	1.5

Table 2
Traffic index characteristics.

Model category	Morning peak	Flat peak	Evening peak	Second peak			
				Moment of emergence	Starting time	Peak value	Duration
Monday mode	7.5–7.9	1.3–2.0	5.6–7.9				
Midweek model	6.3–7.6	1.9–2.4	7.4–8.2	After morning peak	09:30–09:45	4.3–7.2	15 min–45 min
Friday mode	6.1–7.7	2.0–3.0	8.2–8.5	After morning peak, after evening peak	09:30–09:45, 20:15–20:30	3.6–5.2	45 min–1 h 45 min
Saturday mode	4.2–6.8	2.7–6.6	7.1–8.2	Before morning peak, before evening peak	08:30–08:45, 14:45–15:00	3.5–6.6	15 min–1 h 15 min
Sunday mode	2.6–3.9	2.3–5.5	6.3–7.6	Before morning peak, before evening peak	08:45–09:00, 14:30–15:00	1.6–3.6	1 h 15 min–3 h

moments), 10:00–12:15 (10 moments), 12:30–14:45 (10 moments), 15:00–17:15 (10 moments), 17:30–19:45 (10 moments), and 20:00–23:00 (13 moments). The index data of the seven time periods were obtained using the ARIMA algorithm, and then the predicted values of the seven time periods were combined into a 1-day prediction index in time order. Let us consider the Monday model data prediction as an example. We used the 9-day data of November 7, 2016, November 21, 2016, November 28, 2017, November 6, 2017, November 13, 2017, November 20, 2017, November 27, 2018, November 5, 2018, and November 12, 2018 to make predictions and compare the accuracy with the actual values.

This experiment was implemented in TensorFlow using the CPU version and Python. The field named “Traffic Index” was read. The output is shown in Fig. 4. Using the data of the Monday mode, we predicted the time series data and drew autocorrelation graphs, as shown in Fig. 5. The ADF test was carried out for seven periods in Monday mode. The test results are shown in Table 3.

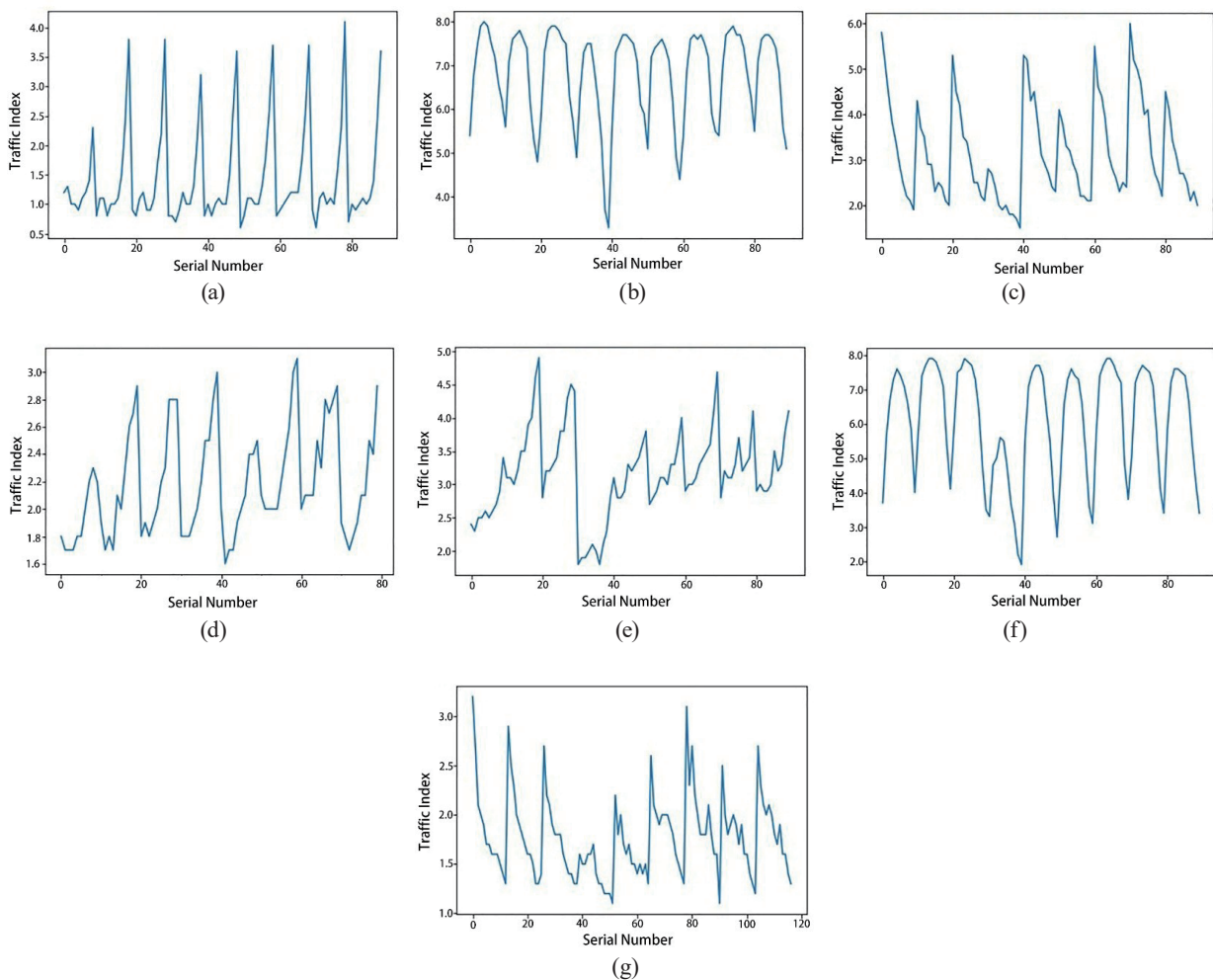


Fig. 4. (Color online) Monday mode data training set for (a) 05:00–07:15, (b) 07:30–09:45, (c) 10:00–12:15, (d) 12:30–14:45, (e) 15:00–17:15, (f) 17:30–19:45, and (g) 20:00–23:00.

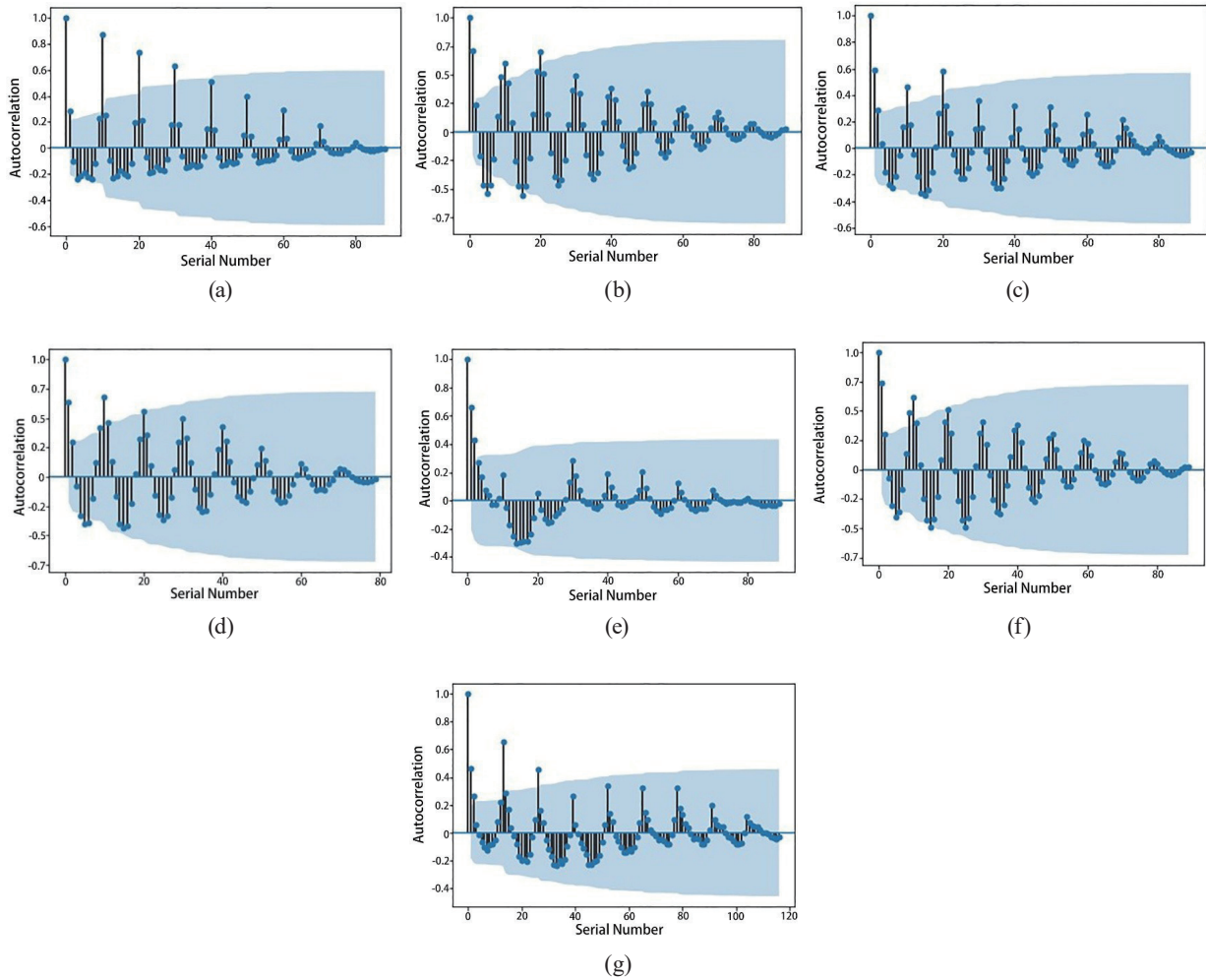


Fig. 5. (Color online) Autocorrelation graph of Monday pattern time series for (a) 05:00–07:15, (b) 07:30–09:45, (c) 10:00–12:15, (d) 12:30–14:45, (e) 15:00–17:15, (f) 17:30–19:45, and (g) 20:00–23:00.

Table 3
ADF test values of each period.

Time	Test statistic	<i>p</i> -value	Lags used	Number of observations used	Critical value (1%)	Critical value (5%)	Critical value (10%)
05:00–07:15	−2.423	0.135	10	78	−3.517	−2.899	−2.587
07:30–09:45	−2.217	0.200	8	81	−3.514	−2.898	−2.586
10:00–12:15	−3.475	0.009	11	78	−3.517	−2.899	−2.586
12:30–14:45	−2.692	0.075	11	68	−3.530	−2.905	−2.590
15:00–17:15	−3.011	0.034	10	79	−3.516	−2.899	−2.587
17:30–19:45	−2.883	0.0474	10	79	−3.516	−2.898	−2.586
20:00–23:00	−1.372	0.596	13	103	−3.495	−2.890	−2.582

From the time series graphs, autocorrelation graphs, and ADF test values, seven sets of time series data were found to be nonstationary data. We performed the differential operation on the seven sets of data (the operation results are shown in Fig. 6), and the autocorrelation graphs

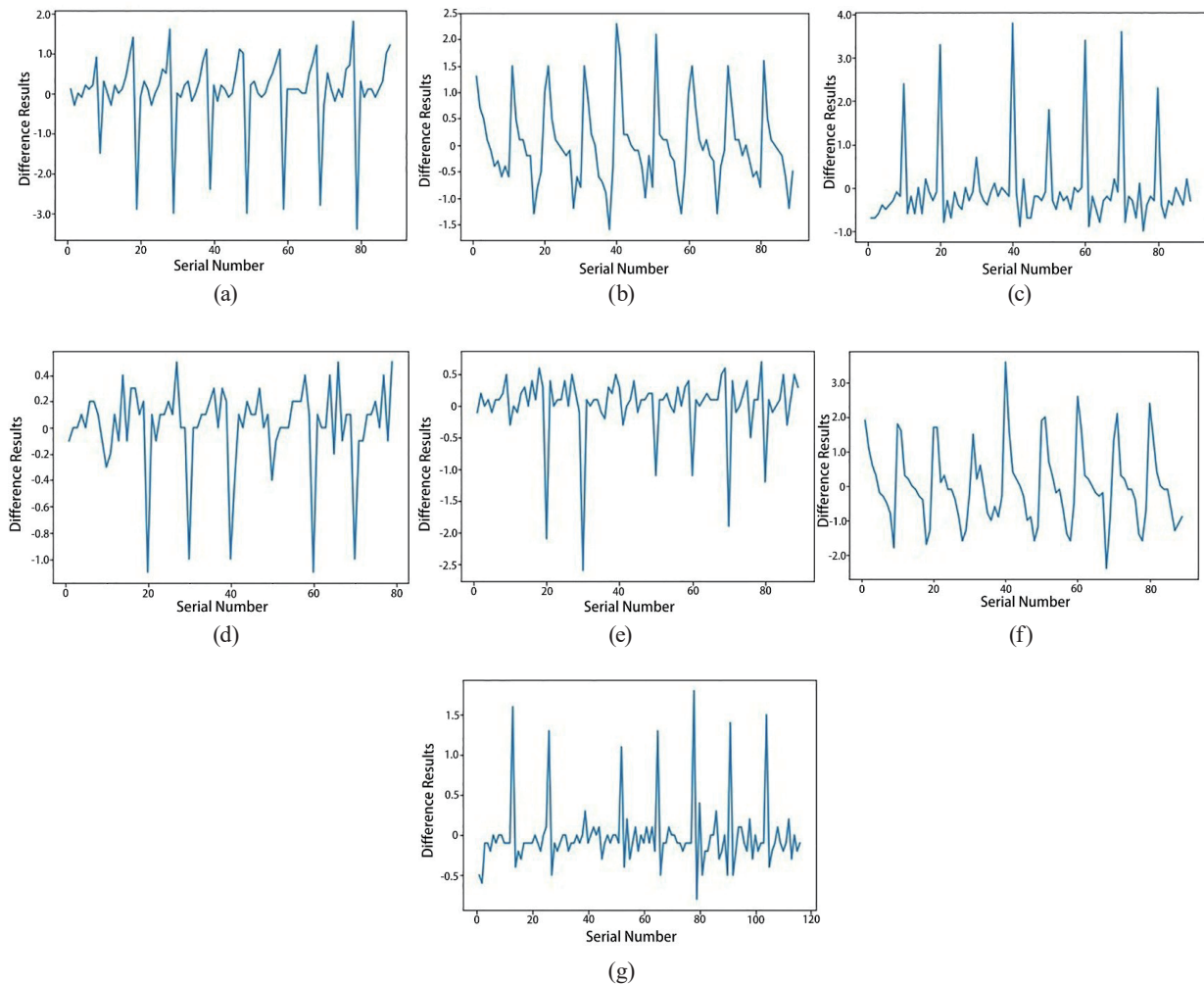


Fig. 6. (Color online) Differential graphs of data in Monday mode time series for (a) 05:00–07:15, (b) 07:30–09:45, (c) 10:00–12:15, (d) 12:30–14:45, (e) 15:00–17:15, (f) 17:30–19:45, and (g) 20:00–23:00.

and partial autocorrelation graphs are respectively shown in Figs. 7 and 8. The ADF test was performed on the time series of the traffic index through the differential calculations. The test results are shown in Table 4.

From the differential index distribution graphs, autocorrelation graphs, partial correlation graphs, and ADF test results, the p value is less than 0.05. This indicates that the nonstationary time series data were converted into a stationary series through the differential operation. Based on the time series data of the steady traffic index, the Bayesian information criterion (BIC) matrix was selected to determine the values of the parameters p and q in each period, as shown in Table 5. We predicted the time series by fitting the ARIMA (prediction results retain three significant digits), as shown in Table 6. Seven sets of time series prediction results were integrated to form a 1-day traffic index prediction result with 1×73 dimensions. A comparison between the predicted result and the actual traffic index is shown in Fig. 9. The overall change of the predicted index is consistent with the real situation, and the predicted result conforms to the characteristics of the Monday pattern.

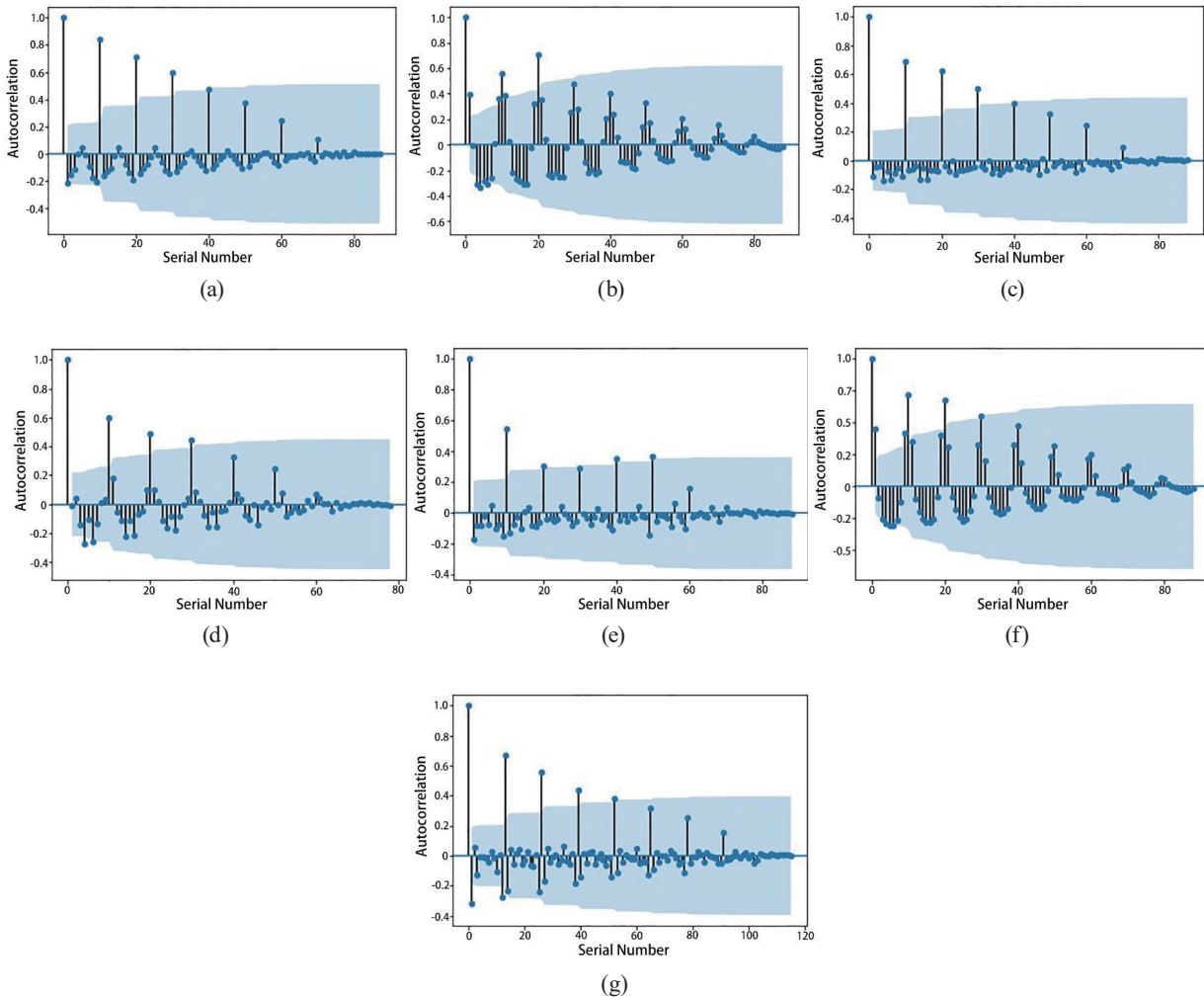


Fig. 7. (Color online) Differential autocorrelation graphs for different time series on Monday mode for (a) 05:00–07:15, (b) 07:30–09:45, (c) 10:00–12:15, (d) 12:30–14:45, (e) 15:00–17:15, (f) 17:30–19:45, and (g) 20:00–23:00.

Table 4
ADF test values after difference.

Time	Test statistic	p -value	Lags used	Number of observations used	Critical value (1%)	Critical value (5%)	Critical value (10%)
05:00–07:15	−5.576	1.433e−06	9	78	−3.517	−2.899	−2.587
07:30–09:45	−7.977	2.678e−12	7	81	−3.514	−2.898	−2.586
10:00–12:15	−2.885	0.047	7	79	−3.515	−2.899	−2.587
12:30–14:45	−2.531	0.008	10	68	−3.530	−2.905	−2.590
15:00–17:15	−2.757	0.035	7	79	−3.516	−2.899	−2.587
17:30–19:45	−2.778	0.011	7	79	−3.516	−2.899	−2.587
20:00–23:00	−4.538	0.0002	12	103	−3.495	−2.890	−2.582

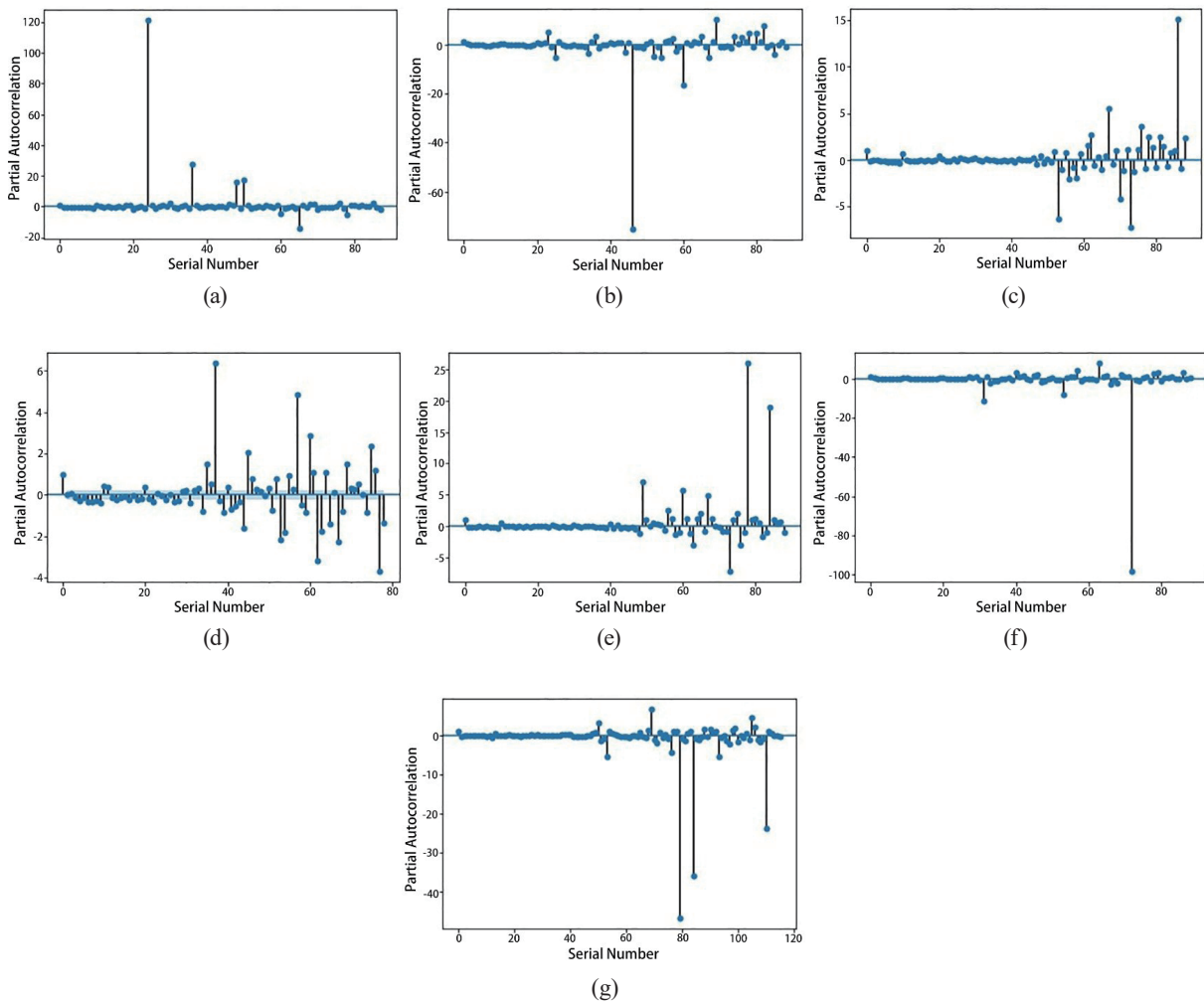


Fig. 8. (Color online) (Continued) Partial difference autocorrelation graphs of Monday pattern time series for (a) 05:00–07:15, (b) 07:30–09:45, (c) 10:00–12:15, (d) 12:30–14:45, (e) 15:00–17:15, (f) 17:30–19:45, and (g) 20:00–23:00.

Table 5
Parameters p and q in each period.

Time	p	q
05:00–07:15	8	5
07:30–09:45	5	2
10:00–12:15	8	5
12:30–14:45	2	3
15:00–17:15	6	3
17:30–19:45	7	8
20:00–23:00	1	1

Table 6
Prediction results of each period.

Time	Prediction results
05:00–07:15	[1.45,0.90,1.21,1.10,1.19,1.27,1.32,1.12,2.59,3.21]
07:30–09:45	[5.50,6.43,7.21,7.65,7.76,7.59,7.09,6.35,5.68,5.40]
10:00–12:15	[4.06,3.79,2.68,2.86,2.78,2.72,2.62,2.52,1.88,2.14]
12:30–14:45	[2.31,2.09,1.97,1.99,2.15,2.38,2.61,2.73,2.72,2.57]
15:00–17:15	[2.88,2.75,2.68,2.63,2.61,2.60,2.59,2.59,3.59,3.60]
17:30–19:45	[5.23,7.15,7.28,7.39,7.30,6.92,6.40,5.57,3.81,3.52]
20:00–23:00	[2.50,2.60,2.66,2.69,1.70,1.71,1.71,1.72,1.72,1.72,1.71,1.71]

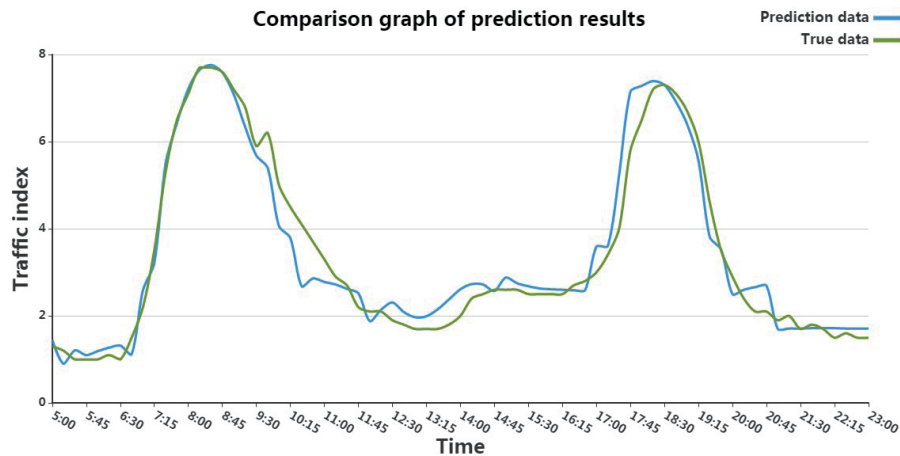


Fig. 9. (Color online) Comparison of prediction results.

5.2 Pattern classification

In this experiment, the data from November 2016, November 2017, and November 1–15, 2018 were used as training sets and the data from November 16–29, 2018 were used as test sets. They were labeled according to their true categories. For example, (1, 0, 0, 0, 0) denoted the Monday mode, (0, 1, 0, 0, 0) denoted the midweek mode, (0, 0, 1, 0, 0) denoted the Friday mode, (0, 0, 0, 1, 0) denoted the Saturday mode, and (0, 0, 0, 0, 1) denoted the Sunday mode. The daily data were supplemented with the numerical value “0” at the first place of the time series data to obtain a 10×10 matrix. The labels were marked at the end of the matrix to distinguish the patterns of each day, forming the input layer of the CNN.

A CNN network was used as a deep learning model, and the experimental process was implemented in TensorFlow by using the CPU version of Python language. Even \times even convolution kernels can lead to the loss of image boundary information and the offset of position information. Therefore, convolution kernels are usually in the form of odd \times odd. The size of the convolution kernel is determined by the size of the input layer. The common sizes include 3×3 , 5×5 , and 7×7 . Increasing the edge length of the convolution kernel will lead to a sharp increase in the amount of computation. The receptive field range of a 7×7 convolution kernel and the amount of calculation were too large, and the difference in calculation results was not obvious in this experiment; the 5×5 convolution kernel had a larger receptive field range than the 3×3 convolution kernel and could extract more feature information. Therefore, we selected a convolution kernel with a size of 5×5 . Based on the operation characteristics of the mean pooling layer and the maximum pooling layer, the mean pooling layer is more inclined to retain the background information, and the maximum pooling layer is better at extracting the texture features of the neighborhood. In this study, we divided the traffic index time series on the basis of the change law, and the values with the most obvious characteristic difference could most effectively represent the change law of the traffic index, so the background information characteristics could be ignored, and the neighborhood maximum value was

explored to describe the characteristic information of the traffic index. The operation mode of the maximum pooling layer met the requirements of this experiment, and 2×2 maximum pooling layers were used to realize local feature compression and extraction in this experiment. A larger pooling layer caused a loss of layer information and reduced the robustness of the extracted features.

The overall framework used in this experimental pattern classification section was composed of the CNN and Softmax classifiers. The CNN framework was composed of two groups of convolution layers: the ReLU activation functions and the maximum pooling layers; the Softmax classifier was composed of the Softmax function classification layers, and the fully connected layer connected the CNN and Softmax classifiers. Because of the small sample size of traffic index data in this experiment, we used two groups of convolution kernels and pooling layers to extract features to avoid overfitting. First, the 10×10 matrix of the input layer was extracted by the 5×5 convolution kernel. After convolution processing, the ReLU activation function was used for the nonlinear conversion to solve the problems of gradient disappearance and overfitting. The result of nonlinear conversion of the activation function was processed by using the 2×2 maximum pooling layers. The second group of the convolution pooling process was the same as the first one. The output layers unified the multidimensional data through the fully connected layer after the two-step processing and used output layers of CNN as the input layer of the Softmax function logistic regression, output the probability of each mode, and took the mode corresponding to the maximum probability as the category of the time series data mode for that day.

The test data identification results are shown in Table 7. We were able to distinguish the Monday mode (November 19, November 22, November 26, and November 27), midweek mode (November 20, November 21, November 28, and November 29), Friday mode (November 16 and November 23), Saturday mode (November 17 and November 24), and Sunday mode (November 18 and November 25) using the CNN. However, it mistakenly classified November 22 and November 27 into the Monday mode, which was inconsistent with the actual situation. The classification accuracy of the test set reached 85.7%.

The prediction results were processed in the same way as above and constructed into a 10×10 matrix as the input layer. Then, the CNN was used to classify the prediction results, and a Monday pattern was obtained, which was consistent with the actual situation. The experimental results show that the ARIMA-CNN model discussed in this paper can predict and classify traffic index time series data. The classification results were used as the judgment criteria,

Table 7
Experimental results of test set.

Categories	Experimental results	Actual conditions
Monday mode	19, 22, 26, 27	19, 26
Midweek mode	20, 21, 28, 29	20, 21, 22, 27, 28, 29
Friday mode	16, 23	16, 23
Saturday mode	17, 24	17, 24
Sunday mode	18, 25	18, 25

and the corresponding model threshold lines of the corresponding categories were invoked as a comparison with the prediction results. The outliers beyond the threshold were the focus of this work. Traffic departments can take abnormal values as the basis for decision-making in traffic control and reduce the incidence of urban traffic congestion.

6. Discussion and Conclusion

To make up for the deficiencies of traditional traffic congestion analysis, we proposed the ARIMA-CNN model to realize the traffic index prediction and classification of traffic index types and to explore the abnormal conditions that occur in traffic to make predictions and enhance early warning systems. In the experiment, the traffic index data of Beijing in 2016–2018 were taken as the research object. The ADF was used to test whether the traffic index data of Beijing formed a nonstationary series, which was then transformed into a stationary series by the differential operation. The traffic index was predicted by ARIMA fitted by the BIC matrix. The CNN algorithm was used to extract the time series pattern characteristics of the traffic index to distinguish the prediction pattern of the traffic index. The experimental results show that the ARIMA-CNN model accurately predicts the traffic index data of Beijing and recognizes the prediction result as a Monday pattern, which matched the actual situation. The model has the ability to generate accurate predictions and pattern recognition of time series data, which can assist the relevant departments of traffic management to conduct traffic management in advance and relieve urban traffic pressure.

In future research, we will consider the impact of traffic emergencies, severe weather, and major events on the weights of the model (combined with the Internet of Things data) to help with predictions and decision-making that reduce the incidence of urban traffic congestion as much as possible.

Acknowledgments

Support from numerous classmates, friends and experts from Beijing University of Civil Engineering and Architecture is gratefully acknowledged. We also would like to thank Beijing Traffic Information Center, who provided all the traffic data.

This research was funded by the National Natural Science Foundation of China (grant nos. 41771413 and 41701473), the Beijing Natural Science Foundation (grant no. 8202013), the China Scholarship Council (grant no. 201909960002), Key Projects of Strategic International Scientific and Technological Innovation Cooperation (grant no. 2018YFE0206100), the Graduate Innovation Project Fund (grant no. PG2020085), and the Research Project of China Construction Education Association (grant no. 2019046).

References

- 1 Y. Pei, S. Biswas, D. S. Fussell, and K. Pingali: *Commun. ACM* **62** (2017) 11. <https://doi.org/10.1145/3363294>
- 2 M. Baptistaa, E. M. P. Henriques, I. P. Medeiros, J. P. Malere, C. L. Nascimento, and H. Prendinger: *Reliab. Eng. Syst. Saf.* **184** (2019) 228. <https://doi.org/10.1016/j.ress.2018.01.017>

- 3 D. Yang: Sol. Energy. **182** (2019) 179. <https://doi.org/10.1016/j.solener.2019.02.044>
- 4 Q. Wang and F. Jiang: Energy **178** (2019) 781. <https://doi.org/10.1016/j.energy.2019.04.115>
- 5 H. Duan, D. Wang, X. Pang, Y. Liu, and S. Zeng: J. Cleaner Prod. **260** (2020) 120929. <https://doi.org/10.1016/j.jclepro.2020.120929>
- 6 S. Fan and L. Chen: IEEE Trans. Power Syst. **21** (2006) 1. <https://doi.org/10.1109/TPWRS.2005.860944>
- 7 A. Kalra and S. Ahmad: Water Resour. Res. **45** (2009) 3. <https://doi.org/10.1029/2008WR006855>
- 8 G. Lin, Y. Chou, and M. Wu: J. Hydrol. **486** (2013) 334. <https://doi.org/10.1016/j.jhydrol.2013.02.012>
- 9 Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu: IET Intel. Transport Syst. **11** (2017) 68. <https://doi.org/10.1049/iet-its.2016.0208>
- 10 X. Ran, Z. Shan, Y. Fang, and C. Lin: Sensors **19** (2019) 861. <https://doi.org/10.3390/s19040861>
- 11 S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman: 2017 Int. Conf. Advances in Computing, Communications and Informatics (ICACCI) (2012) 1643–1647.
- 12 G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer: Lect. Notes Comput. Sci. **2888** (2003) 986. https://doi.org/10.1007/978-3-540-39964-3_62
- 13 A. Ahmad and L. Dey: Data Knowl. Eng. **63** (2007) 503. <https://doi.org/10.1016/j.datak.2007.03.016>
- 14 L. Zhu, C. Lu, and Y. Sun: IEEE Trans. Power Syst. **31** (2016) 1430. <https://doi.org/10.1109/TPWRS.2015.2413895>
- 15 K. B. Ensor: J. Am. Stat. Assoc. **97** (2002) 656. <https://doi.org/10.1198/jasa.2002.s477>
- 16 M. Murat, I. Malinowska, M. Gos, and J. Krzyszczak: Int. Agrophys. **32** (2018) 253. <https://doi.org/10.1515/intag-2017-0007>
- 17 J. W. Lichstein, T. R. Simons, S. A. Shriner, and K. E. Franzreb: Ecol. Monogr. **72** (2002) 445. [https://doi.org/10.1890/0012-9615\(2002\)072\[0445:SAAAMI\]2.0.CO;2](https://doi.org/10.1890/0012-9615(2002)072[0445:SAAAMI]2.0.CO;2)
- 18 H. Cecotti: Pattern Recognit. Lett. **32** (2001) 1145. <https://doi.org/10.1016/j.patrec.2011.02.022>
- 19 J. Chu, Z. Guo, and L. Leng: IEEE Access **6** (2018) 19959. <https://doi.org/10.1109/ACCESS.2018.2815149>
- 20 Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner: Proc. IEEE. **86** (1998) 2278. <https://doi.org/10.1109/5.726791>
- 21 J. Schmidhuber: Neural Networks **61** (2015) 85. <https://doi.org/10.1016/j.neunet.2014.09.003>
- 22 Y. Zhang, D. Zhao, J. Sun, G. Zou, and W. Li: Neural Process. Letters. **43** (2015) 389. <https://doi.org/10.1007/s11063-015-9420-y>
- 23 X. Deng, Y. Zhang, S. Yang, P. Tan, L. Chang, Y. Yuan, and H. Wang: IEEE Trans. Image Process. **27** (2018) 1888. <https://doi.org/10.1109/TIP.2017.2779600>
- 24 S. Totaro, A. Hussain, and S. Scardapane: Neurocomputing **381** (2020) 177. <https://doi.org/10.1016/j.neucom.2019.10.084>

About the Authors



Jian Lu received his B.E. degree in surveying and mapping engineering from North China Institute of Science and Technology. He is currently studying for an M.S. in surveying and mapping engineering at Beijing University of Civil Engineering and Architecture. His research interests focus on intelligent transportation and geographic information systems. (lujian_0719@foxmail.com)



Xuedong Zhang is an associate professor and master's tutor of Beijing University of Civil Engineering and Architecture. He graduated from China University of Mining and Technology (Beijing) in 2012 with a doctorate degree in engineering and is currently engaged in research on geographical information systems and urban remote sensing. He has published more than 30 related papers and won four provincial and ministerial awards. (zhangxuedong@bucea.edu.cn)



Zhijie Xu is an associate professor and master's tutor of Beijing University of Civil Engineering and Architecture. She mainly teaches advanced mathematics, probability theory, and mathematical statistics courses. Her research areas include deep learning, computer vision, machine learning, and data mining. In addition to teaching, she also presides over the National Natural Science Foundation Youth Fund project, publishing monographs, and participating in the compilation of teaching materials. She has published many papers and national invention patents. (xuzhijie@bucea.edu.cn)



Jianqin Zhang graduated with a doctorate degree from the Institute of Remote Sensing Applications of the Chinese Academy of Sciences. In 2009, he became a postdoctoral fellow of the Institute of Geographical Sciences and Natural Resources Research of the Chinese Academy of Sciences. He is currently a member of the Beijing Intelligent Transportation Association, a member of the Virtual Digital Environment Professional Committee of the International Digital Earth Association Central Committee, and a member of the China Geographic Information Industry Association Theory and Method Working Committee. (zjq2001_54@163.com)



Jingjing Wang is a senior engineer and vice director of the Beijing Municipal Transportation Operations Coordination Center. She has been engaged in the field of intelligent transportation for more than 10 years, with rich experience in intelligent transportation planning, scientific research and engineering project construction, comprehensive traffic operation monitoring services, and so on. She was responsible for and also involved in a number of national research projects, major projects, and research topics at the provincial and ministerial levels, won four ministerial and provincial-level awards, published more than 10 papers, applied for three invention patents, and obtained more than 10 software copyrights as well. (wangjingjing@jtw.beijing.gov.cn)



Lizeng Mao received his Ph.D. from China Agricultural University in 2008. He is currently a senior engineer in Beijing Municipal Transportation Operations Coordination Center. His research interests focus on urban traffic management and intelligent transportation system. He has published more than 50 academic papers related to transportation in various journals, and he is a member of the China Intelligent Transportation Systems Association and Beijing Traffic Engineering Association. (maolizeng@163.com)



Lipeng Jia received his B.E. degree in geographic remote sensing science and technology from Shandong University of Science and Technology. He is currently studying for an M.S. in surveying and mapping engineering at Beijing University of Civil Engineering and Architecture. His research interests focus on intelligent transportation and geographic information systems. (1310087449@qq.com)



Zhuohang Li received her B.E. degree in geographic information systems from Beijing University of Civil Engineering and Architecture. She is currently studying for an M.S. in surveying and mapping engineering at Beijing University of Civil Engineering and Architecture. Her research interests focus on intelligent transportation and geographic information systems. (823400490@qq.com)