Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6682216, 9 pages
https://doi.org/10.1155/2020/6682216

Research Article

A Short-Term Traffic Flow Reliability Prediction Method considering Traffic Safety

Shaoqian Li,[1] Zhenyuan Zhang,[1] Yang Liu,[1] and Zixia Qin[2]

[1] *Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China*
[2] *School of Design and Arts, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Zhenyuan Zhang; zzhenyuan@cqjtu.edu.cn

Received 4 November 2020; Revised 30 November 2020; Accepted 30 November 2020; Published 10 December 2020

Academic Editor: Yong Chen

With the rapid development and application of intelligent traffic systems, traffic flow prediction has attracted an increasing amount of attention. Accurate and timely traffic flow information is of great significance to improve the safety of transportation. To improve the prediction accuracy of the backward-propagation neural network (BPNN) prediction model, which easily falls into local optimal solutions, this paper proposes an adaptive differential evolution (DE) algorithm-optimized BPNN (DE-BPNN) model for a short-term traffic flow prediction. First, by the mutation, crossover, and selection operations of the DE algorithm, the initial weights and biases of the BPNN are optimized. Then, the initial weights and biases obtained by the aforementioned preoptimization are used to train the BPNN, thereby obtaining the optimal weights and biases. Finally, the trained BPNN is utilized to predict the real-time traffic flow. The experimental results show that the accuracy of the DE-BPNN model is improved about 7.36% as compared with that of the BPNN model. The DE-BPNN is superior to the performance of three classical models for short-term traffic flow prediction.

1. Introduction

In recent years, with the development of traffic detection technology, big data technology, and data mining technology, accurate and real-time traffic flow operation data and traffic accident data are easy to collect [1]. By studying the changing characteristics of traffic flow before and after traffic accidents, the traffic safety status is analyzed, evaluated, and forewarned according to the collected real-time traffic flow data. Real-time and dynamic release of early warning information can adjust and control traffic flow parameters in time, greatly reduce the occurrence of traffic accidents and the degree of accident damage, and thus improve the operation efficiency of the expressway network. Among them, the intelligent prediction of traffic flow plays a key role in various technologies. Short-term traffic flow forecasting is the most valuable practice in traffic application, and it is the foundation and basis for the realization of advanced traffic management system and traffic information service system. The accuracy of short-term traffic flow prediction directly affects the effects of traffic flow guidance and traffic control, which is of great significance for maintaining traffic safety. Real-time and accurate traffic flow prediction is the premise and key to the realization of both traffic flow guidance systems and traffic control systems [2].

As a matter of fact, short-term traffic flow prediction largely relies on the historical and real-time traffic data collected through various sensors (e.g., induction coils, radar, cameras, mobile global positioning systems, and social media) to build corresponding models and algorithms. The fundamental principle of short-term traffic flow prediction is as follows: first, a rational prediction model is built by a dedicated structure and parameters based on a certain amount of sensor data such as the historical traffic flow, vehicle density, and vehicle speed; then, the prediction model is trained by the corresponding learning algorithm based on the collected data to obtain a set of optimal solutions; finally, the traffic sensor data to be determined are fed back into the trained model to predict the future traffic flow. By principle, existing short-term traffic flow prediction methods are broadly classified into parametric methods, nonparametric methods, and simulation approaches [3]. Parametric methods principally include time series models, Kalman filtering, autoregression, and exponential smoothing. Nonparametric methods include *K*-nearest neighbor methods, support vector machines (SVMs), and artificial neural networks. Simulation approaches predict traffic flow using existing traffic simulation tools. These classical short-time traffic flow prediction methods offer favourable results for theoretical analysis and simulation. Unfortunately, their application in practical engineering scenarios is greatly limited due to the explosive growth of traffic big data.

Accordingly, to solve the data explosion problem arising from the explosive growth of traffic data, many short-term traffic flow prediction methods based on improving the parameters of the aforementioned classical models have been developed. For instance, based on a classical parametric prediction method, the seasonal autoregressive integrated moving average (ARIMA) model, Williams and Hoel [4] built a short-term traffic flow prediction model by considering the impact of seasonal factors on road traffic flow. Furthermore, considering the influencing factors of affect spatiotemporal correlations such as the road network topology and time-varying speed, Duan et al. [5] proposed a spatiotemporal model based on the Space-Time Autoregressive Integrated Moving Average (STARIMA) model, which further enhances the accuracy of short-term traffic flow prediction. However, it is difficult to achieve high traffic flow prediction accuracy with the limited small samples. Therefore, Kumar [6] assumed a linear traffic flow and attached great importance to the temporal correlation of traffic flow at a particular location with a relatively stable traffic flow, thereby proposing a Kalman filter-based prediction scheme, which improves the prediction accuracy using small samples. However, such methods neglect the impacts of complex and changeable actual traffic environment challenges, such as spatiotemporal interaction and coupling.

To solve this problem, building new short-term traffic flow prediction models by combining models based on nonparametric prediction methods is considered to be a solution for short-term traffic flow prediction [7, 8]. Duo et al. [9] optimized the parameters of an SVM and built a short-term traffic flow prediction model by decomposing the traffic flow sequence into different frequency components and then introducing the crossover and mutation factors of the genetic algorithm into particle swarm optimization (PSO). Dai et al. [10] proposed a gated recurrent unit-(GRU-) based short-term traffic flow prediction model based on an analysis of the spatiotemporal characteristics of traffic flow data. Chen et al. [11] attempted to build a number of prediction models with different time delays to propose the least squares support vector regression (LSSVR) based short-term traffic flow prediction model and to achieve a favourable prediction performance. Zhao et al. [12] proposed a hybrid model by combing *K*-nearest neighbor (KNN) with support vector regression (SVR), imitating the KNN search mechanism to rebuild a historical traffic flow sequence Unfortunately, since the road network traffic system is affected by uncertainties such as the road traffic environment, weather conditions, and pedestrians, the actual traffic flow data are evidently nonlinear, time-varying, and susceptible to random noise. Therefore, the above traffic flow prediction methods are not suitable for short-term traffic flow prediction in complex conditions because they are still limited by dedicated model parameters, low prediction accuracy, and poor generalization. So, exploring more effective methods to achieve higher short-term traffic flow prediction accuracy has become a great concern for traffic researchers.

Recently, deep learning-based methods, such as backward-propagation neural networks (BPNNs), have been successfully applied to various tasks in traffic flow prediction. Some scholars have introduced artificial neural networks with many hidden layers to build short-term traffic flow prediction models [13–15], which achieve better prediction performance. However, BPNNs have two obvious shortcomings, including a high involvement in local optimal solutions and a low convergence rate. Meanwhile, these models lack the interpretability of the learning process. Hence, how to optimize the structural parameters of a BPNN and building a practicable short-term traffic flow prediction model is the main focus of this article.

For the above problems, based on the influence of traffic volume and traffic safety, this paper proposes an adaptive differential evolution (DE) algorithm-optimized BPNN (DE-BPNN) model for short-term traffic flow prediction. First, the DE algorithm is used for heuristic random optimization of the group difference of the BP parameters based on a brief description of the BPNN to make up for the random defects of the BPNN in terms of the initial weight and bias selection. Second, to accelerate the convergence rate in short-term traffic flow prediction, the BPNN is combined with the DE algorithm to build a novel short-term traffic prediction model for global optimization and generalization of short-time traffic flow. Finally, simulation verification is performed for the proposed algorithms and models using the standard data set collected by the Caltrans Performance Measurement System (PeMS), USA. The simulation results show that the proposed algorithm has a better learning ability and global optimization performance compared to conventional algorithms such as ARIMA, wavelet neural networks (WNNs), and BPNNs.

2. Traffic Flow Prediction Modelling Based on DE-BPNN

In general, traffic flow prediction can be classified by the prediction period into long-term prediction, medium-term prediction, and short-term prediction. In fact, once travellers learn the evolution trend of short-term traffic flow in real-time, they can change their routes for fast, convenient, and comfortable travel. Therefore, travellers extremely concern about short-term traffic flow prediction. In actual traffic environments, the 5- to 30-minute traffic flow evolution trend is chosen as the time range for short-term traffic flow prediction. To ensure the prediction accuracy, the traffic flow data sequence observed within *n* identical time intervals at an observation point in the traffic network is assumed to be

$\{x_i\}, i = 1, 2, \ldots, n$, and the predicted traffic flow of a certain period in the future is y. With a rational traffic flow prediction model, it is possible to observe the traffic flow data sequence for accurate prediction within a short time, thereby providing an effective decision-making basis for travellers to choose their travel routes. This paper adopts the DE-BPNN-based model for the optimization and improvement of accuracy. For the simplicity of analysis and the integrity of the overall frame structure, the basic structure of the BPNN-based short-term traffic flow prediction model is introduced first.

2.1. Overview of the BPNN-Based Short-Term Traffic Flow Prediction Model.

It is well known that the traffic system is a large, complex, nonlinear, time-varying, and stochastic system. BPNN can identify complex nonlinear systems and constantly adjust the parameters based on a large number of collected data sequences. Moreover, it can approximate any nonlinear continuous function with an arbitrary precision through the deep data fusion of parallel structures and the data processing capability of self-learning. Such properties help to remarkably reduce the computing workload of online prediction. Therefore, BPNN-based methods are widely applied in the field of short-term traffic flow prediction [16, 17]. Generally, the BPNN structure includes an input layer, a hidden layer, and an output layer. Each layer is connected by weights and bias. The weights and bias value range is typically $[-1, 1]$. As shown in Figure 1, a neuron model contains an input layer with n nodes, an intermediate layer with m nodes, and an output layer with 1 node. In short-term traffic flow prediction, the processing procedure basic execution process consists of the forward propagation of traffic information and the backward propagation of error, as shown in Figure 2.

It is assumed that x_i represents the traffic flow of an observation point in the traffic network in the i^{th} time interval; the input is $(x_1, x_2, \ldots, x_n)^T$, and the output is y.

$$\begin{cases} \text{net}_j = \sum_{i=1}^{n} \omega_{ij} * x_i + \theta_j, \\ y_j = f_j(\text{net}_j), \\ \text{net} = \sum_{j=1}^{m} v_j * y_j + \theta, \\ y = f(\text{net}), \end{cases} \quad j = 1, 2, \ldots m, \tag{1}$$

where w_{ij} and v_j are the connection weight between the input-hidden layers and hidden-output layers, respectively; θ_j and θ are the biases of the hidden layer and output layer, respectively; $f_j(\cdot)$ and $f(\cdot)$ are the activation functions of the hidden layer and output layer, respectively.

The BPNN weights and biases can generate three matrices and one bias, including, the weight matrix W from the input layer to the hidden layer, the weight matrix V from the hidden layer to the output layer, and the bias matrix T of the hidden layer and the output layer bias θ. Each matrix is represented as follows:

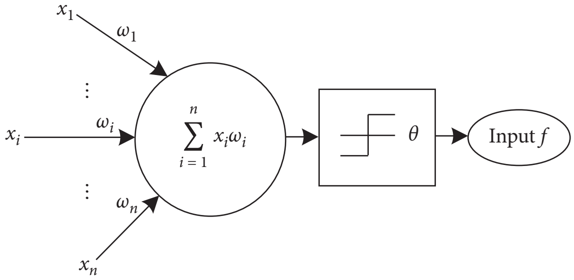

Figure 1: Neuron model structure.

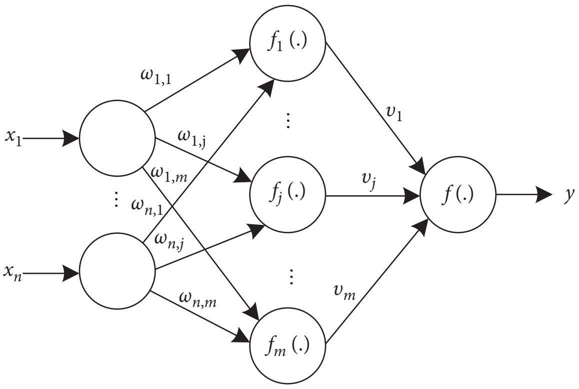

Figure 2: Single hidden layer BPNN structure diagram.

$$\begin{aligned} W &= \begin{bmatrix} \omega_{1,1} & \omega_{1,2} & \cdots & \omega_{1,n} \\ \omega_{2,1} & \omega_{2,2} & \cdots & \omega_{2,n} \\ \vdots & \vdots & & \vdots \\ \omega_{m,1} & \omega_{m,2} & \cdots & \omega_{m,n} \end{bmatrix}, \\ V &= [v_1, v_2, \ldots, v_m], \\ T &= [\theta_1, \theta_2, \ldots, \theta_m]^T. \end{aligned} \tag{2}$$

Compared with classical traffic flow prediction algorithms, the BPNN has obvious superiorities, such as higher prediction accuracy and noise robustness. However, such a black box-like learning mode still faces several challenges [7]. Firstly, since the training process utilizes the current real-time data, the trained network may be no longer applicable when the traffic flow changes. Secondly, the connection weights and bais of each layer in BPNN are set randomly, which may make the training process fall into local minimization. To alleviate these challenges, this paper relies on the DE algorithm to optimize BPNN parameters, resulting in faster convergence, simpler implementation, and higher prediction accuracy.

2.2. Improvement of the Model by DE Algorithm-Based BPNN Parameter Optimization.

As a group difference-based heuristic global search algorithm, the DE algorithm optimizes the distribution of the weights and biases for each layer in BPNN through real number encoding. During the iteration process, the optimal weights and biases are chromosomes obtained by the assistance of the DE algorithm. In the end, the local optimization of the BPNN-based traffic flow prediction model ultimately leads to the global optimal solution. Conventionally, the DE algorithm employs a different strategy for mutation operations, that is, the difference vector between individuals in the population to perturb individuals to achieve individual variation. The mutation method of the DE algorithm improves the search capacity by using the characteristics of the population distribution effectively.

In fact, the DE optimization process adopts population initialization, mutation, crossover, and selection strategies to map matrices, including W, V, T, and θ, into the chromosome. The mapping relationship is expressed as follows:

$$\left\{\omega_{1\ ,\ 1}, \omega_{1\ ,\ 2}, \ldots, \omega_{n,m}, \upsilon_1, \upsilon_2, \ldots, \upsilon_m, \theta_1, \theta_2, \ldots, \theta_m, \theta\right\}, \tag{3}$$

where the set of mappings is D, which represents the number of dimensions of a variable. The initial variables are calculated and assigned based on equation (4), and the initial population $\{z_i(0) | z_{i.j}^L \leq z_{i.j}(0) \leq z_{i.j}^U, i = 1, 2, \ldots, \mathrm{Np}, j = 1, 2, \ldots, D\}$ is generated randomly as given in the following equation.

$$z_{i,j}(0) = z_{i.j}^L + \mathrm{rand}(0,1)\left(z_{i.j}^U - z_{i.j}^L\right), \tag{4}$$

where $z_i(0)$ represents the i^{th} individual of the 0^{th} generation in the population; $z_{i.j}(0)$ represents the j^{th} "gene" (number of dimensions) of the i^{th} individual in the 0^{th} generation in the population; $z_{i.j}^L$ and $z_{i.j}^U$, respectively, represent the minimum and maximum numbers of dimensions of the individual; Np represents the population size; and $\mathrm{rand}(0,1)$ is a random number uniformly distributed in the interval $(0,1)$.

A mutation operation is performed to achieve individual mutation through a differential strategy. Three different individuals are randomly selected from Np numbers of individuals; two of them are scaled by the vector difference, and another vector is added thereto, that is,

$$v_i(g+1) = z_{r1}(g) + F\left(z_{r2}(g) - z_{r3}(g)\right), \tag{5}$$

where $z_i(g)$ represents the i^{th} individual in the g^{th} generation population; $r1, r2, r3$, and target vector i are different from each other; and F is a scale factor, which has been assigned a value between $[0,2]$ and used to control the scaling of differential variables [18, 19].

The crossover operation is performed between individuals for the g^{th} generation population $\{z_i(g)\}$ and the intermediate $\{v_i(g+1)\}$

$$u_{i.j}(g+1) = \begin{cases} v_{i.j}(g+1), if\, \mathrm{rand}(0,1) \leq \mathrm{CR}\ \mathrm{or}\, j = j_{\mathrm{rand}}, \\ z_{i.j}(g), \mathrm{otherwise}, \end{cases} \tag{6}$$

where CR is the crossover factor and j_{rand} is a random integer in $[1, 2, \ldots, D]$. To ensure at least one "gene" in the intermediate individual is passed to the next generation, the $j_{\mathrm{rand}}^{\mathrm{th}}$ gene of each individual is passed in the first crossover "gene" operation. $z_i(g)$ or $v_i(g+1)$ is chosen as the allele of $u_i(g+1)$, which depends on the CR probability for subsequent crossover operations.

The selection strategy focuses on the population selection after the crossover operation in the differential algorithm. Based on the complexity of the actual traffic environment, the DE algorithm employs the greedy algorithm to choose the individuals inputting to the next generation of the traffic flow population; that is,

$$z_i(g+1) = \begin{cases} u_i(g+1), & f\left(u_i(g+1)\right) \leq f\left(x_i(g)\right), \\ x_i(g), & \text{otherwise.} \end{cases} \tag{7}$$

For the DE algorithm, the mutation, crossover, and selection operations are continuously performed through equations (5)–(7) until meeting the conditions. Thus, the DE-BPNN parameter optimization is achieved. Note that, the scale and crossover factors of the DE algorithm are fixed values based on experience. In actual traffic flow prediction, this algorithm requires a wider search range to avoid becoming trapped in local optimal solutions in the early stage, while it requires a narrower search range to prevent the algorithm from missing the extreme points in the later stage. Therefore, to conform to the dynamics of short-term traffic flow, it is necessary to improve the method of determining the scale and crossover factors.

2.3. Method of Determining the Dynamics of Adaptive Scale and Crossover Factors for DE Algorithm.

To ensure the accuracy and effectiveness of short-term traffic flow prediction, the established adaptive scale factors need to guarantee the following characteristics: as the number of iterations increases, the mutation rate should gradually decrease; at the beginning of an iteration, a larger scale factor should be selected to increase the diversity of the traffic flow population; a smaller scale factor should be selected in the later stage to preserve the superior individuals of the traffic flow population. Based on these considerations, adaptive scale factors are generated with the following equation:

$$F = F_0 * \widehat{2}\, e^{1-\left(G_m/G_m+(1-G)\right)}, \tag{8}$$

where F_0 represents the initial scale factor, G_m is the maximum number of iterations, and G stands for the current number of iterations.

Similarly to the scale factor, as the number of iterations increases, the crossover rate changes dynamically. The larger crossover factor at the initial stage ensures the global traffic flow state mutation. The smaller crossover rate in the later period is more focused on local traffic state convergence. Hence, the design adaptive crossover factor is shown in the following equation:

$$\mathrm{CR} = \mathrm{CR}_{\max} - \left(\frac{G\left(\mathrm{CR}_{\max} - \mathrm{CR}_{\min}\right)}{G_m}\right), \tag{9}$$

where $\mathrm{CR}_{\max}$ is the maximum value of the crossover parameter and $\mathrm{CR}_{\min}$ represents the minimum value of the crossover parameter.

In regard to traffic flow prediction, because the DE algorithm optimizes the BP parameters, it can prevent the

BPNN from being trapped in local optimal solutions and improve the accuracy of the BPNN-based traffic flow algorithm.

3. Design of the Short-Term Traffic Flow Prediction Algorithm Based on Adaptive DE-BPNN

The preceding section introduces the BPNN-based short-term traffic prediction model and the DE algorithm, as well as the method for determining the dynamic parameter factors in the DE algorithm. The basic procedure of the hybrid short-term traffic flow prediction model algorithm is given below, as shown in Figure 3.

Step 1: initialize the population. N_p populations are randomly initialized based on equation (4). Each individual has D dimensions, each of which represents a parameter in a neural network.

Step 2: the error between the neural network output and actual values is defined as the population-dependent fitness function. The fitness of each individual is calculated, and the minimum fitness value is the global minimum; the globally optimal individual is updated. The fitness function can be represented by the mean square error (MSE) or the root mean square error (RMSE) as given in the following equation:

$$\text{MSE} = \frac{1}{N_{ts}} \sum_{i=1}^{N_{ts}} \left(\tilde{y}_i - \hat{y}_i\right)^2,$$
$$\text{RMSE} = \sqrt{\frac{1}{N_{ts}} \sum_{i=1}^{N_{ts}} \left(\tilde{y}_i - \hat{y}_i\right)^2}, \tag{10}$$

where N_{ts} represents the number of trained samples; $\tilde{y}_i$ is the actual value of the predicted traffic flow; and $\hat{y}_i$ stands for the value of the actual traffic flow.

Step 3: the next-generation individual $x_i(g+1)$ is obtained based on the DE mutation, crossover, and selection operations.

Step 4: step 3 is repeated until the next-generation population is obtained.

Step 5: determine whether or not the termination condition (the global minimum meets the predefined accuracy requirements, or the maximum number of iterations is reached) is met; if yes, the iteration will be stopped with the optimal individuals as parameters of the neural network; otherwise, go to the next step.

Step 6: if $g = g + 1$; go back to Step 2.

Step 7: enter the test set and perform prediction with the trained network.

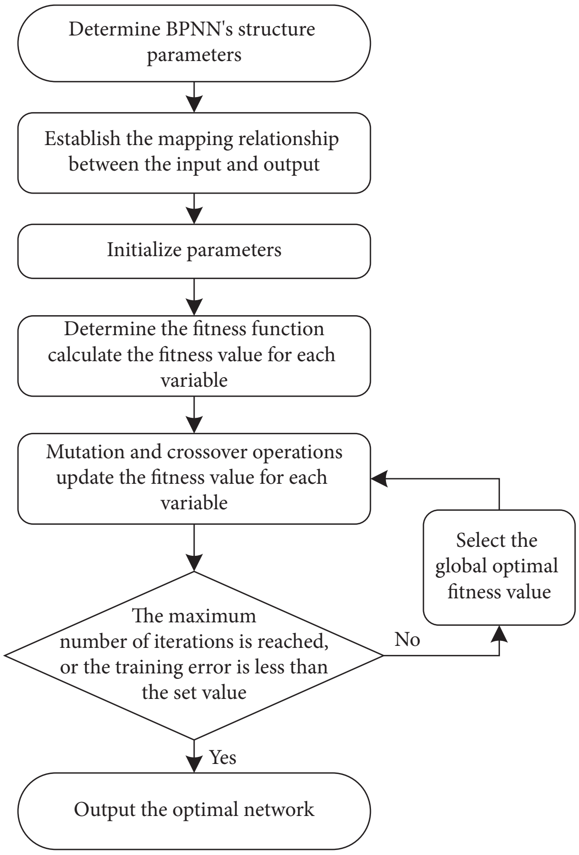
Determine BPNN's structure parameters
Establish the mapping relationship between the input and output
Initialize parameters
Determine the fitness function calculate the fitness value for each variable
Mutation and crossover operations update the fitness value for each variable
Select the global optimal fitness value
The maximum number of iterations is reached, or the training error is less than the set value
No
Yes
Output the optimal network

Figure 3: DE-BPNN algorithm flowchart.

4. Simulation Experiment

4.1. Experiment Conditions. To verify the performance of the proposed short-term traffic flow prediction model, the PeMS data set, one of the most commonly used data set in short-term traffic flow prediction [20], was selected. The data acquired from one road segment were chosen for detection; the data collected by these detectors were summarized once in every 5 minutes, and the traffic flow was summarized for three one-way lanes. Then, the data were subjected to preprocessing, including the removal of redundant data, the correction of erroneous data, and normalization. Figure 4 shows the traffic flow at a detection point along a freeway overtime during the week. It can be seen that the traffic flow on weekdays tends to be consistent. To more closely assess the similarity of weekdays, the daily traffic flow was compared in the same plane. As shown in Figure 5, the morning and evening rush hours are almost at the same time on the weekdays, which reflects the consistency of travel patterns. Hence, to ensure the accuracy of the prediction, weekends and weekdays are distinguished, and the experimental study was conducted on weekday traffic flow data collected from May 2 to May 6, 2018. The data collected on May 2 and 3, 2018, were used as the training set, while the data acquired on May 4 were used as the prediction set.

4.2. Assessment Indicators. To evaluate the effectiveness of the DE-BPNN model and some conventional models, the four most commonly used performance indicators were selected for regression problems: the mean absolute error (MAE), the MSE, the mean absolute percentage error

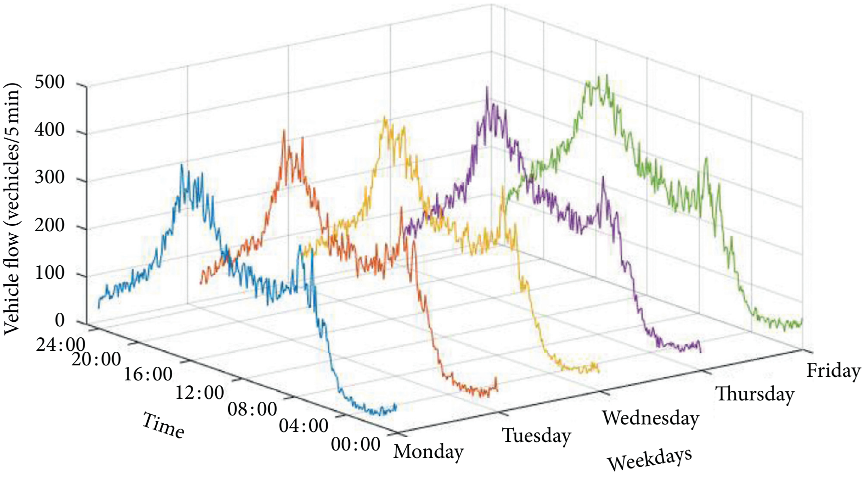

Figure 4: Correlation of time series on weekdays.

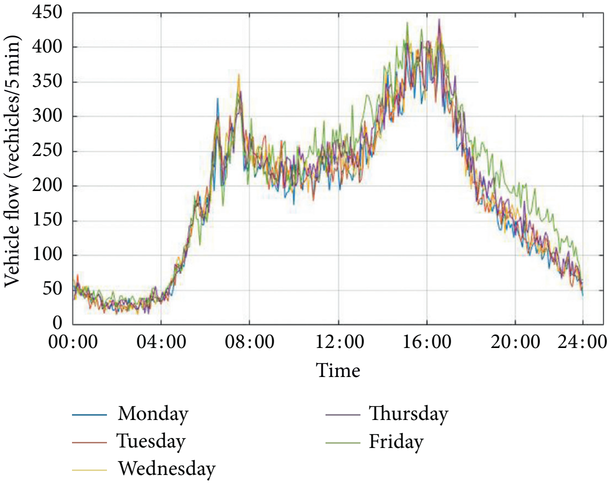

Figure 5: Typical traffic flow pattern on weekdays.

(MAPE), and the mean square percent error (MSPE). All indicators are defined as follows: y_t and $\widehat{y}_t$ represent the detection value and the model prediction value of traffic flow, respectively [21, 22].

(1) Mean absolute error (MAE): it can avoid the problem of mutual cancellation of errors, so it can well reflect the actual situation of predicted value errors.

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - \widehat{y}_t\right|. \tag{11}$$

(2) Mean square error (MSE): it is a measure reflecting the difference between the estimated quantity and the estimated quantity, which can evaluate the change degree of the data. The smaller the MSE value, the better the accuracy of the prediction model in describing the experimental data:

$$\mathrm{MSE} = \frac{1}{n}\sqrt{\sum_{t=1}^{n}\left(y_t - \widehat{y}_t\right)^2}. \tag{12}$$

(3) Mean absolute percentage error (MAPE): it is used as a statistical indicator to measure the accuracy of prediction. A smaller MAPE indicates a better model effect.

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \widehat{y}_t}{y_t}\right|. \tag{13}$$

(4) Mean square percent error (MSPE): it is used to test the degree of model fitting. The smaller the MSPE is, the better the fitting degree is, and the model can be accepted.

$$\mathrm{MSPE} = \frac{1}{n}\sqrt{\sum_{t=1}^{n}\left(\frac{y_t - \hat{y}_t}{y_t}\right)^2}. \tag{14}$$

4.3. Model Parameters. The experiment used the BPNN-based method with a single hidden layer ($m - 2m + 1 - 1$). As shown in Figure 2, the input data were set as $(x_1, x_2, \ldots, x_n)^T$, and the output result was y[23, 24]. The parameters of the BPNN model are as follows: the maximum number of trainings is 1,000, the training error target is 0.001, the learning rate is 0.01, the activation function of the hidden layer is tansig: $f(x) = (2/(1 + e^{-2x})) - 1$, the activation function of the output layer is purelin: $f(x) = x$, and the training function is trainlm. The parameter settings for the DE algorithm are as follows: the population size is $N_p = 10$, the number of iterations is $g = 100$, the results of several trial calculations determine the mutation factor is $F_0 = 0.9$, and the crossover factor $\mathrm{CR}_{\min} = 0.1$ and $\mathrm{CR}_{\max} = 0.7$. The experiment was performed on a PC with an Intel i7 2.4 GHZ CPU and an 8 GB RAM; the algorithm was written in the MATLAB R2018a environment.

4.4. Interpretation of Results

4.4.1. Model Prediction Results. In this paper, the traffic flow data observed at an observation point in the road network on May 4 were analyzed. The prediction results are shown in Figure 6. It can be seen from Figure 6 that the DE-BPNN model yields excellent prediction results. The predicted values at each time point are basically consistent with the actual values, so the traffic flow trend throughout the day is excellently predicted, and the trend of the change in the traffic flow is accurately identified.

4.4.2. Comparative Analysis of Models. Theoretically, the adaptive DE-BPNN model proposed in this paper offers higher convergence rates and smaller prediction errors compared with the use of the BPNN alone. In the early stage, the DE algorithm can effectively avoid the local extremum problem and offer fast convergence and optimization. Furthermore, this algorithm can find the optimal initial parameters of the BPNN during training and continuously optimize the BPNN, and the parameter values are more accurate than the initial values randomly generated by the neural network, thereby enhancing the prediction accuracy. To compare the predictive performance of the DE-BPNN model, three classical prediction models were selected for comparison, including the ARIMA-based, WNN-based, and BPNN-based models. The results of the short-term traffic flow predictions performed with the ARIMA, BPNN, and WNN models are given below. The data in Figures 7–9 include the actual values, the predicted values, and the prediction errors, including the emergence of morning and evening peaks.

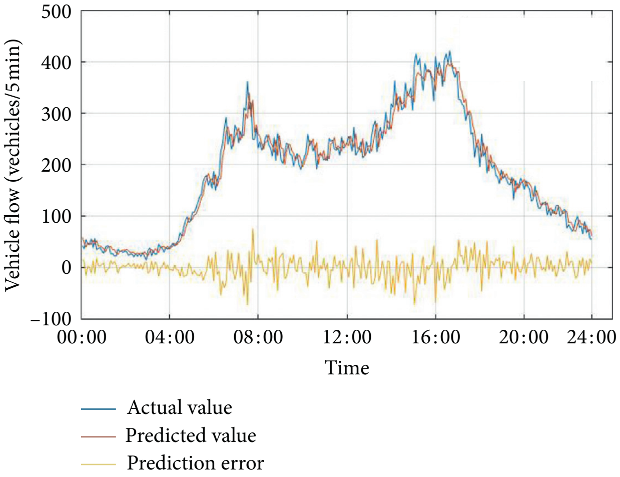

FIGURE 6: Traffic flow prediction results of the DE-BPNN model.

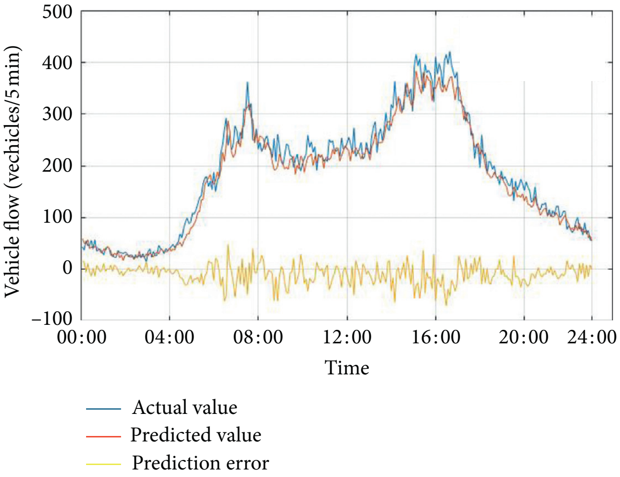

FIGURE 7: Traffic flow prediction results with the ARIMA model.

For a more intuitive comparison of the model prediction results, the four performance evaluation indicators are used to evaluate the four prediction models.

The performances of the ARIMA, WNN, BPNN, and DE-BPNN models were compared. We used the same data set in all cases. Table 1 shows the prediction result of the 5 min freeway traffic flow verification data set. It should be noted that we only used the traffic flow data as the input for prediction without considering engineering factors related to the traffic flow, such as weather conditions, accidents, and other traffic flow parameters (density and speed). As shown in Table 2, the MAE values of DE-BPNN model decreases 49.04%, 8.16%, and 7.36% as compared with those of the ARIMA, WNN, and BPNN models, respectively; the MSE values of DE-BPNN model decreases 44.97%, 5.88%, and 6.66% as compared with those of the ARIMA, WNN, and BPNN models, respectively; the MAPE values of the DE-BPNN decreases 33.43%, 19.68%, and 18.55% as compared with those of the ARIMA, WNN, and BPNN models, respectively; and the MSPE values decreases 23.58%, 29.85%,

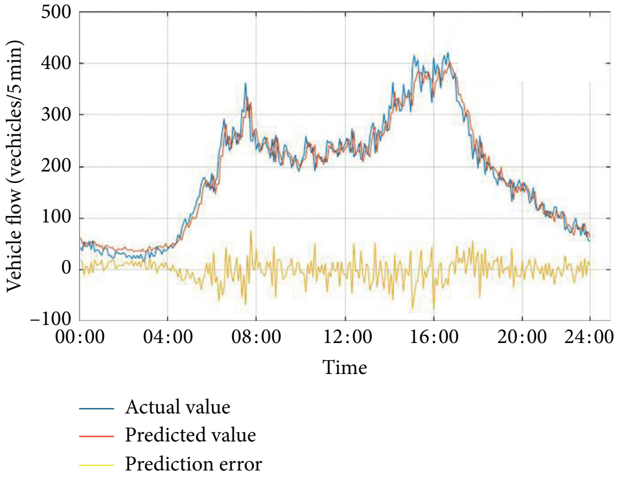

Figure 8: Traffic flow prediction results with the WNN model.

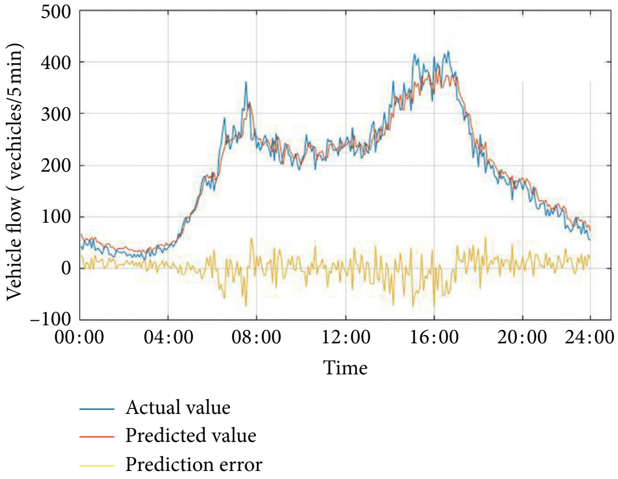

Figure 9: Traffic flow prediction results with the BPNN model.

Table 1: Prediction performances of the ARIMA, WNN, BPNN, and DE-BPNN models.

Model	MAE	MSE	MAPE	MSPE
ARIMA	31.7049	2.3084	0.1735	0.0123
WNN	17.5927	1.3496	0.1438	0.0134
BPNN	17.4396	1.3609	0.1418	0.0130
DE-BPNN	16.1563	1.2702	0.1155	0.0094

Table 2: Percentage improvement in the prediction results with the DE-BPNN model compared with the ARIMA, WNN, and BPNN models.

Model	MAE (%)	MSE (%)	MAPE (%)	MSPE (%)
ARIMA	49.04	44.97	33.43	23.58
WNN	8.16	5.88	19.68	29.85
BPNN	7.36	6.66	18.55	27.69

and 27.69% as compared with those of the ARIMA, WNN, and BPNN models, respectively.

5. Conclusions

Short-term traffic flow prediction is of great social and economic significance for reducing traffic safety hazards, reducing traffic accidents, providing safe and efficient experience for highway travellers, and improving highway traffic and transportation efficiency. By integrating intelligent optimization algorithm theory with machine learning methods, this paper proposes a DE-BPNN model for short-term traffic flow prediction. Restricted by a variety of internal and external factors, the actual traffic system exhibits strong nonlinearity and uncertainty. The BPNN is suitable for any nonlinear fitting; furthermore, to avoid being trapped in local extrema during the conventional BPNN training process, a difference-based heuristic random search DE algorithm is used for global preoptimization, and then, the weights and biases obtained from the DE algorithm are used to train the BPNN, thus improving the prediction accuracy. The results show that the DE-BPNN prediction model effectively improves the prediction accuracy of short-term traffic flow. Compared with the ARIMA, WNN, and BPNN models, the DE-BPNN model leads to lower values for the MAE, MSE, MAPE, and MSPE error evaluation indicators. The research scope can be expanded, and more complex road network data can be used for experiments in the future. Traffic flows may be affected by weather, traffic accidents, and other factors. How to use such auxiliary information to improve the prediction accuracy will also be a focus in future studies.

Data Availability

The data used to support the findings of this study are available at http://pems.dot.ca.gov.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61573076), Science Technology Research Program of Chongqing Municipal Education Commission (Grant no. KJZD-K201800701), Program of Chongqing Innovation and Entrepreneurship for Returned Overseas Scholars of P.R. China (Grant no. cx2018110), Chongqing Natural Science Foundation (Grant no. cstc2020jcyj-msxmX0797), the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant no. KJQN202000717), and the Innovation Foundation of Chongqing Postgraduate Education (Grant no. CYS20282).

References

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] D. Huang, Z. Deng, S. Wan, B. Mi, and Y. Liu, "Identification and prediction of urban traffic congestion via cyber-physical link optimization," *IEEE Access*, vol. 6, pp. 63268–63278, 2018.

[3] Y. Lv, Y. Duan, W. Kang, and Z. Li, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[4] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[5] P. Duan, G. Mao, W. Yue et al., "A unified STARIMA based model for short-term traffic flow prediction," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1652–1657, IEEE, Maui, Hawaii, USA, 2018.

[6] S. V. Kumar, "Traffic flow prediction using kalman filtering technique," *Procedia Engineering*, vol. 187, pp. 582–587, 2017.

[7] J. Zhang, S. Zhao, Y. Wang, and X. Zhu, "Improved social emotion optimization algorithm for short-term traffic flow forecasting based on back-propagation neural network," *Journal of Shanghai Jiaotong University (Science)*, vol. 24, no. 2, pp. 209–219, 2019.

[8] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[9] M. Duo, Y. Qi, G. Lina et al., "A short-term traffic flow prediction model based on EMD and GPSO-SVM," in *Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pp. 2554–2558, IEEE, Chongqing, China, 2017.

[10] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU," *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[11] X. Chen, X. Cai, J. Liang, and Q. Liu, "Ensemble learning multiple LSSVR with improved harmony search algorithm for short-term traffic flow forecasting," *IEEE Access*, vol. 6, pp. 9347–9357, 2018.

[12] L. Zhao, D. Wei, Y. Dong-Mei et al., "Short-term traffic flow forecasting based on combination of k-nearest neighbor and support vector regression," *Journal of Highway & Transportation Research & Development*, vol. 12, no. 1, pp. 89–96, 2018.

[13] Y. Ma, Z. Zhang, and A. Ihler, "Multi-lane short-term traffic forecasting with convolutional LSTM network," *IEEE Access*, vol. 8, pp. 34629–34643, 2020.

[14] Y. Gu, W. Lu, X. Xu et al., "An improved bayesian combination model for short-term traffic prediction with deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1332–1342, 2019.

[15] W. Cheng and P. Feng, "Network traffic prediction algorithm research based on PSO-BP neural network," in *Proceedings of the 2015 International Conference on Intelligent Systems Research and Mechatronics Engineering*, Atlantis Press, Zhengzhou, China, 2015.

[16] S. A. Zargari, S. Z. Siabil, A. H. Alavi et al., "A computational intelligence-based approach for short-term traffic flow prediction," *Expert Systems*, vol. 29, no. 2, pp. 124–142, 2012.

[17] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction for a non urban highway using artificial neural network," *Procedia—Social and Behavioral Sciences*, vol. 104, no. 2, pp. 755–764, 2013.

[18] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[19] J. Ye, J. Zhao, K. Ye et al., "How to build a graph-based deep learning architecture in traffic domain: a survey," 2020, https://arxiv.org/abs/2005.11691.

[20] Caltrans, "Performance measurement system (PeMS)," 2020, http://pems.dot.ca.gov.

[21] D. Huang, Z. Deng, L. Zhao et al., "A short-term traffic flow forecasting method based on markov chain and grey verhulst model," in *Proceedings of the 2017 6th Data Driven Control and Learning Systems (DDCLS)*, pp. 606–610, IEEE, Chongqing, China, 2017.

[22] D. Huang, Z. Deng, and B. Mi, "A new synergistic forecasting method for short-term traffic flow with event-triggered strong fluctuation," *Journal of Control Science and Engineering*, vol. 2018, Article ID 4570493, 8 pages, 2018.

[23] B. L. Smith and M. J. Demetsky, "Traffic flow forecasting: comparison of modeling approaches," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 261–266, 1997.

[24] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: overview of objectives and methods," *Transport Reviews*, vol. 24, no. 5, pp. 533–557, 2004.