

## Research Article

# Short-Term Traffic Flow Forecasting Model Based on GA-TCN

**Rongji Zhang** , **Feng Sun** , **Ziwen Song** , **Xiaolin Wang** , **Yingcui Du** ,  
and **Shulong Dong** 

*Department of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255000, China*

Correspondence should be addressed to Feng Sun; sunfeng@sdut.edu.cn

Received 27 August 2021; Accepted 24 November 2021; Published 17 December 2021

Academic Editor: Georgia Aifadopoulou

Copyright © 2021 Rongji Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic flow forecasting is the key to an intelligent transportation system (ITS). Currently, the short-term traffic flow forecasting methods based on deep learning need to be further improved in terms of accuracy and computational efficiency. Therefore, a short-term traffic flow forecasting model GA-TCN based on genetic algorithm (GA) optimized time convolutional neural network (TCN) is proposed in this paper. The prediction error was considered as the fitness value and the genetic algorithm was used to optimize the filters, kernel size, batch size, and dilations hyperparameters of the temporal convolutional neural network to determine the optimal fitness prediction model. Finally, the model was tested using the public dataset PEMS. The results showed that the average absolute error of the proposed GA-TCN decreased by 34.09%, 22.42%, and 26.33% compared with LSTM, GRU, and TCN in working days, while the average absolute error of the GA-TCN decreased by 24.42%, 2.33%, and 3.92% in weekend days, respectively. The results indicate that the model proposed in this paper has a better adaptability and higher prediction accuracy in short-term traffic flow forecasting compared with the existing models. The proposed model can provide important support for the formulation of a dynamic traffic control scheme.

## 1. Introduction

Urban road resources are limited in time and space, and the continuous growth of motor vehicle ownership increases the pressure of urban traffic. Meanwhile, the similarity of residents' choice of travel route and travel time also aggravates the traffic congestion. Problems including urban traffic accidents, traffic congestion, and environmental pollution have been widely concerned by the governments and the traffic personnel. In order to improve the efficiency of traffic operation, many countries are committed to developing intelligent transportation systems based on communication, sensors, computers, and other technologies [1, 2]. The traffic flow data is important for intelligent transportation system analysis. The historical traffic flow data can be used to predict the future traffic flow, and the traffic management department can take measures to intervene in traffic in advance in order to ease traffic congestion and reduce traffic accidents. Numerous studies have proposed and optimized traffic flow forecasting methods, extracted traffic flow information,

mined potential laws of traffic flow data, and provided theoretical support for intelligent transportation [3].

Most of the early research on traffic flow forecasting are based on linear theoretical models, such as the historical average model (HA), the automatic regression moving average model (ARIMA) [4] and its variants [5, 6], and the Markov chain (MC) [7]. These models assume that the conditional variance of time series is static, which does not exist in real traffic conditions. Later, a variety of data-driven traffic flow forecasting algorithms were developed based on machine learning methods, such as Bayesian network [8], K-nearest neighbor [9], and support vector machine (SVM) [10]. However, these algorithms and their learning methods cannot adapt to dynamic traffic, because they cannot capture the nonlinear characteristics of complex traffic flows, and cannot deal with a large number of structured and unstructured traffic flow datasets.

In recent years, the real-time performance, accuracy, and reliability of traffic data have been significantly improved due to the rapid development of big data and intelligent

transportation technology and the continuous improvement of traffic facilities [11–13]. Deep neural network models with high dimensional data processing and nonlinear data feature mining capabilities have gained significant attention. The prediction model based on deep learning has gradually become a mainstream trend of traffic flow forecasting [14, 15]. Convolutional neural network (CNN) [16] and recurrent neural network (RNN) [17] are two popular networks that are widely used by researchers [18]. Since the cyclic mechanism can perform the same task of each element in the sequence, the RNN is more suitable for sequential processing over time. Therefore, the RNN is widely used in the temporal characteristics of traffic flow. However, the RNN is susceptible to the disappearance of gradient, and it is difficult to capture the long-term temporal characteristics. In order to solve the problem of RNN's long-term dependence on time, Ma et al. proposed a long-term short-term memory network (LSTM), compared it with the traditional prediction methods, and verified the accuracy and generalization of the LSTM prediction model [19]. Considering the temporal and spatial correlation of traffic flow, Zhao et al. established a new traffic prediction model based on an LSTM network through multiple memory cells [20]. In order to reduce the computational complexity, Wu et al. proposed the gated recursive cell network (GRU) based on the LSTM and reduced the three gate functions in the LSTM to two [21]. The performance of the GRU was slightly better due to the simpler gating structure and relatively low computational complexity. The structure of these neural network models uses continuous feedback between the input sequence and the time step to naturally combine the time dependence and show better performance.

The performance of a single neural network model is limited. In order to further improve the adaptability of the prediction model, the LSTM, the GRU, and the other methods have been combined to form a composite model in recent years [22]. Wu et al. used the LSTM to extract the characteristics of the time structure, combined with the ResNet to optimize the overall model, and reduced the occurrence of gradient disappearance or explosion in network degradation [23]. Sun et al. proposed a GRU neural network model, called SSGRU, based on the linear regression coefficient for road network traffic [24].

Although the RNN and the improved models based on the RNN have made substantial progress in traffic flow forecasting, these models still have the problems of low accuracy and low training efficiency. Therefore, to capture the time dependence of traffic flow data more efficiently and improve the applicability of the short-term traffic flow forecasting model, this paper proposes a traffic flow forecasting model based on time convolution neural network optimization of genetic algorithm. Taking the prediction error as the applicability, a genetic algorithm is used to determine the optimal hyperparameters combination of the TCN neural network to improve the prediction accuracy of the TCN neural network.

The rest of this paper is organized as follows. Section 2 introduces time convolutional neural networks. In Section 3, a GA-TCN model based on a genetic algorithm optimized

time convolutional neural network is proposed. The proposed model is verified using examples in Section 4. Finally, Section 5 concludes the paper.

## 2. Time Convolutional Neural Networks

The time convolutional neural network (TCN) [25] derived from a CNN can directly use the powerful features of convolution to extract the features across time steps. Therefore, the TCN shows better performance than both the LSTM and the GRU in many sequential tasks. The main structure of a TCN can be divided into causal convolution for sequence and dilated convolution and residual modules for historical data memory (see Figure 1).

*2.1. Causal Convolution.* Causal convolution is a strict time-constrained model in which the value at time  $t$  depends only on the value at time  $t$  at the next level and before it.

Filter  $F = (f_1, f_2, \dots, f_k)$ , the sequence  $X = (x_1, x_2, \dots, x_k)$ , and the causal convolution at  $x_t$  is

$$(F * X)(x_t) = \sum_{k=1}^K f_k x_{t-K+k}. \quad (1)$$

*2.2. Dilated Convolution.* Dilated convolution allows the input of convolution to have interval sampling, and the sampling rate is controlled by dilations. The higher the level is, the larger the size of  $d$  will be, which can make the size of the effective window grow exponentially with the number of layers.

Filter  $F = (f_1, f_2, \dots, f_k)$ , the sequence  $X = (x_1, x_2, \dots, x_k)$ , and at  $x_t$ , the dilated convolution with the dilation  $d$  is

$$(F_d * X)(x_t) = \sum_{k=1}^K f_k x_{t-(K-k)d}. \quad (2)$$

*2.3. Residual Module.* The training becomes extremely difficult with the increase of network depth mainly because in the process of network training based on stochastic gradient descent, the multilayer backpropagation of error signals can easily cause the phenomenon of "gradient dispersion" or "gradient explosion." A residual network (ResNet) [26] can well solve the problem of difficult training caused by network depth.

In the TCN, the simple connections between the layers are replaced by the residual structures as shown in Figure 2. At the same time,  $x$  is transformed by  $1 \times 1$  Conv to solve the problem that  $x$  and  $F(x)$  cannot be added directly due to different channel numbers.

## 3. GA-TCN Prediction Model

*3.1. Model Structure.* A TCN based on a convolutional neural network can extract the time features of traffic flow across time steps. In this paper, the TCN is selected as the traffic flow forecasting network mainly due to its simple

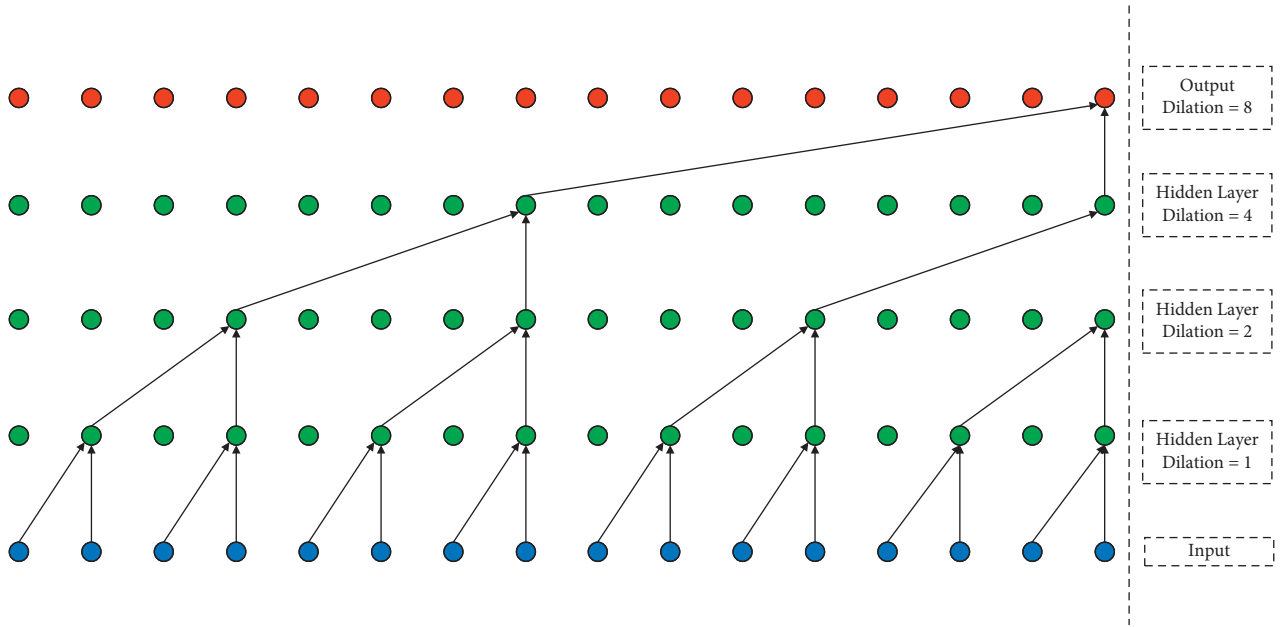


FIGURE 1: Time convolutional neural network structure.

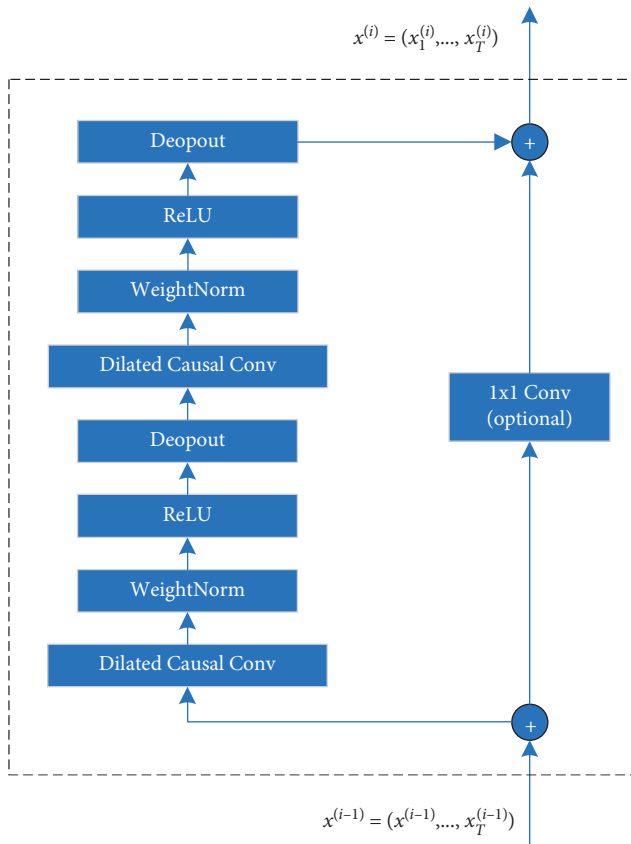


FIGURE 2: Residual connection structure.

structure and strong performance for sequence modeling tasks. The prediction accuracy of the TCN is closely related to the value of the hyperparameter. The filters, kernel size, batch size, and dilations need to be optimized in the search space to determine the best hyperparameters combination.

This paper uses the genetic algorithm to optimize the TCN parameters. The traffic flow time series data is used as the input, the prediction error is used as the fitness value, and the prediction of traffic flow of the next stage is considered as the output matrix. The time iterative convolution neural network adaptive weight adjustment model is used to obtain the optimal solution of search space. The structure of the GA-TCN traffic flow forecasting model is shown in Figure 3.

**3.2. GA Optimization Module.** A genetic algorithm (GA) is a computational model of the biological evolution process that simulates the natural selection and genetic mechanism of Darwinian evolution [27]. The GA is a method to search for the optimal solution by simulating the natural evolution process. It transforms the target problem into the process of biological evolution and generates new populations through crossover, mutation, replication, and other operations and eliminate the solutions with low fitness. The traditional GA is modified with the combination of a time convolutional neural network. The main steps of the optimization process of the modified genetic algorithm are as follows.

- (1) In population initialization, each chromosome is set to the same length. If it fails to meet the length requirements, it is filled with zero. The coding form of the chromosome does not adopt binary form but only exchanges the elements between different genes.
- (2) The mean absolute error (MAE) of the TCN is used as the fitness function:

$$H = \frac{1}{n} \sum_{i=1}^n |y_t - y_p| \quad (3)$$

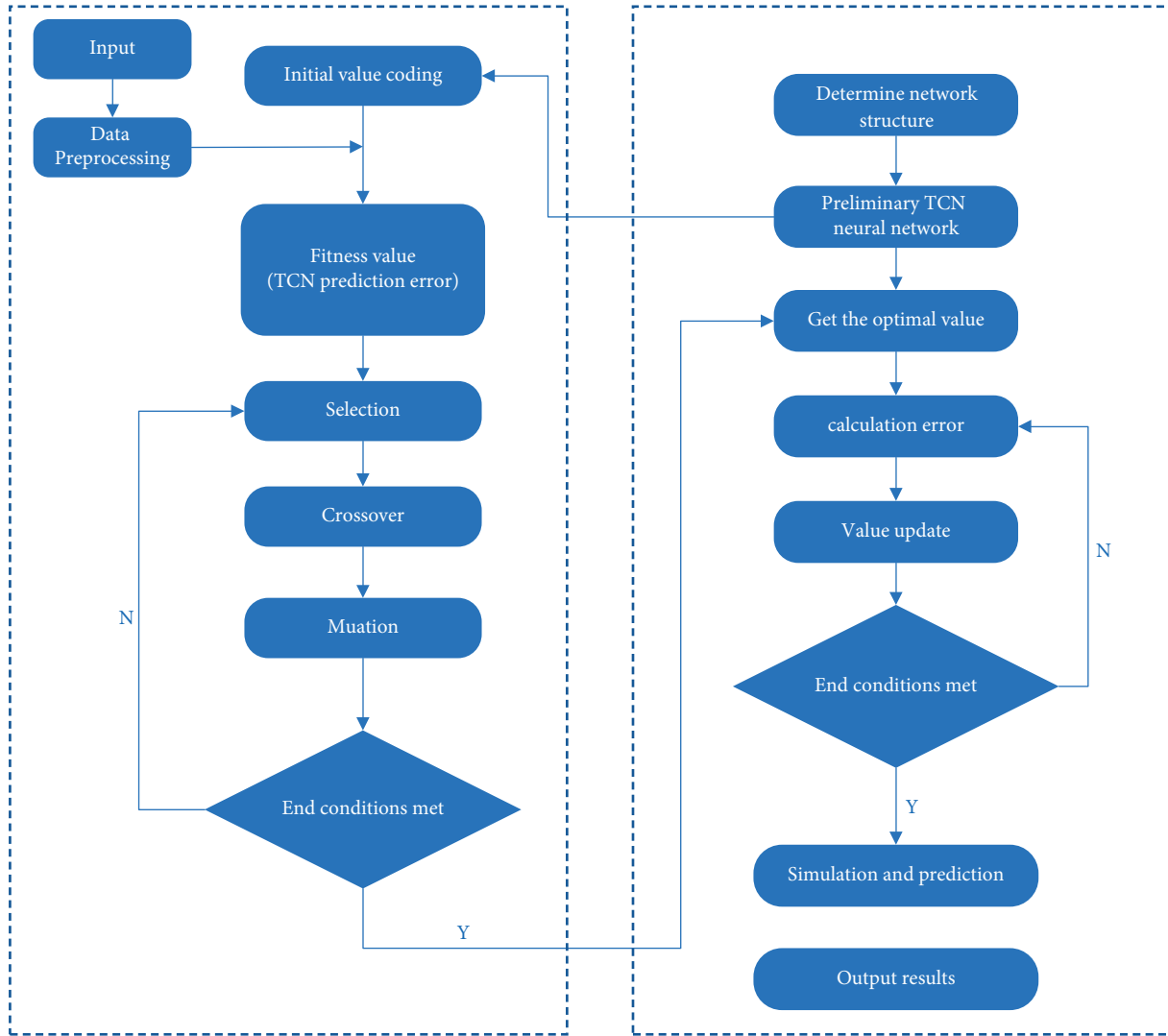


FIGURE 3: Structural diagram of the GA-TCN traffic flow forecasting model.

- (3) The individual of the solution is selected for crossover and mutation operations. The modification of the crossover function consists of two steps. First, identify the positions on the two chromosomes (A and B) that need to be exchanged, and then traverse the genes on the two chromosomes at these positions. If the gene at this location on any chromosome is zero or the gene to be swapped is related to dilations layers, cancel the swap at this location. Generally, the dilation increases by a multiple of 2. Therefore, only filters, kernel size, batch size, and the layers of dilations are mutated.
- (4) If the fitness function target value reaches the optimal value, move to the next step; otherwise, return to step 3.
- (5) Obtain the fitness target value and the best parameters.
- (6) Calculate the MAE of pre-diction based on the best parameters.
- (7) If the number of population iterations is satisfied, the calculation stops, and then the global optimal

hyperparameters combination [filters, kernel size, batch size, and dilations] of the TCN is obtained. Otherwise, return to step 6.

**3.3. TCN Prediction Module.** The TCN prediction module mainly uses one-dimensional causal convolution and a filter  $f$  of width  $k$ . According to the input traffic flow data ( $X = x_{n-1}, x_{n-2}, \dots, x_{n-m}$ ), the convolution operation is performed to obtain A sequence [A B] of length  $m-k+1$  and with the same number of channels:

$$[A B] = f * X_{\text{test}}, \quad (4)$$

where  $*$  represents the convolution operation.

The Dropout setting allows a random selection of neurons to be inactivated to prevent training overfitting and speed up model convergence. The Dropout coefficient is set to 0.1 in the TCN prediction module.

The Adam optimizer is used in the TCN prediction module to optimize the parameters of the model to reduce the loss. Adam is an adaptive optimization algorithm that

performs one-step optimization on a random objective function. It combines the advantages of the AdaGrad and the RMSProp optimizers.

ReLU represents a linear integer function and is used as the activation function of the TCN:

$$\widehat{S}_t^{j,l} = g(W^1 S_{t-s}^{j,l-1} + W^2 S_t^{j,l-1} + b). \quad (5)$$

After adding the residual, the activation function becomes

$$S_t^{j,l} = S_t^{j,l-1} + V\widehat{S}_t^{j,l} + e, \quad (6)$$

where  $W^1$  and  $W^2$  are the weight matrices and  $b$  is the bias vector.

The valid information in  $A$  is extracted by activating the function  $S_t^{j,l}(B)$  and made accessible at the next level. The final output is  $O$ :

$$O = P \odot S_t^{j,l}(B), \quad (7)$$

where  $\odot$  represents the Hadamard product.

## 4. Experiments

**4.1. Data Description.** The experiment in this paper used the public dataset PEMS, which is a real-time collection of California highway traffic data by the Caltrans Performance Measurement System. PEMS data was collected every 30 seconds to form the final data every 5 minutes. The PEMS dataset used in the experiment were collected from a station on highway I05, as shown in Figure 4. The time range was 00:00:00–23:59:59 from May 1<sup>st</sup> to June 30<sup>th</sup> in 2014, with a total of 61 days and 17568 records.

The main conditions of the PEMS dataset are shown in Table 1. The main data used in the experiment were time-stamp and total flow.

Through descriptive statistical analysis of traffic data, distribution histograms and cumulative frequency graphs were drawn, as shown in Figure 5. We can see that the flow is mainly distributed within 20–360. Among them, the proportion in the range of 20–60 and 240–300 is relatively high, and the proportion in the range of 60–240 and 300–360 is relatively low.

**4.2. Data Preprocessing.** Due to equipment failure, noise interference, improper storage, human error, and other emergencies, the data cannot be collected 100%, and there are missing and outlier values. Therefore, it is necessary to preprocess the original data before establishing the prediction model. From the descriptive statistical analysis of the original data, it can be found that there are no large outliers in the data, and only some 0 values exist. In the experiment, we used 0 as the missing value. The sensitivity of the tree model to missing values is low, and it is suitable for the case of moderate or large data. Therefore, the random forest was used to fill in the missing values in this experiment.

The traffic flow data is a nonstationary random sequence. In the experiment, minimum and maximum normalization

was adopted to map the data to the interval  $[0,1]$ . The normalization is defined as

$$z = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (8)$$

where  $z$  is the normalized data,  $x$  is the original data, and  $x_{\min}$  and  $x_{\max}$  are the minimum and maximum values of  $x$ , respectively.

There were obvious differences in the characteristics of sunrise travel between working days and weekend days. Therefore, the data was divided into working days and weekend days. There were 43 working days and 18 weekend days. The data of the last working day and last weekend day were used as the test set, and the rest of the data was used as the training set.

### 4.3. Experiment Settings

**4.3.1. Baseline Methods.** In order to better verify the prediction effect of the model proposed in this paper, the following three baseline methods are added in the experiment for comparison.

The first method is LSTM. It is special RNN that adopts a threshold structure to transform the RNN, solves the problems of gradient disappearance and gradient explosion in regression prediction of long time series of RNN, and can learn long dependence.

The second method is GRU. It is a variant of RNN. The LSTM has three gate functions: input gate, forgetting gate, and output gate, whereas in the GRU model, there are only update and reset gates.

The final method is TCN. Time convolutional neural network is a new algorithm based on CNN that can solve time series prediction.

**4.3.2. Evaluation Metrics.** In this experiment, the root mean square error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE) are used as the evaluation indexes of the model:

$$\begin{aligned} \text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^n (y_t - y_p)^2}, \\ \text{MAE} &= \frac{1}{n} \sum_{i=1}^n |y_t - y_p|, \\ \text{MAPE} &= \frac{1}{n} \sum_{i=1}^n \left| \frac{y_t - y_p}{y_t} \right|, \end{aligned} \quad (9)$$

where  $y_t$  and  $y_p$  are the true and the predicted values of the traffic volume, respectively, and  $n$  is the number of samples.

**4.4. Parameter Optimal Value.** The search range of filters, kernel size, batch size, and dilations layer is set to  $[5-30]$ ,  $[2-10]$ ,  $[8, 16, \dots, 128, 256]$ , and  $[2-8]$ , respectively. The optimal combination of TCN is determined to be

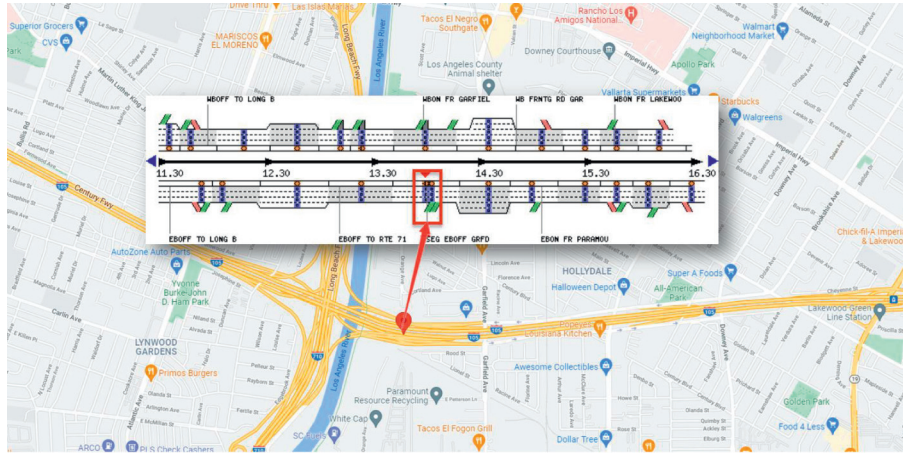


FIGURE 4: The station of data collection.

TABLE 1: The main conditions of the PEMS dataset.

Timestamp	Station	Freeway	Direction of travel	Station length	Total flow	Avg. occupancy	Avg. speed
2014/5/1 0:00	716453	105	E	0.305	59	0.0105	68.5
2014/5/1 0:05	716453	105	E	0.305	60	0.0104	68.6
2014/5/1 0:10	716453	105	E	0.305	59	0.0107	68.8
2014/5/1 0:15	716453	105	E	0.305	57	0.0103	68.7
2014/5/1 0:20	716453	105	E	0.305	53	0.0098	68.6
...	...	...	...	...	...	...	...

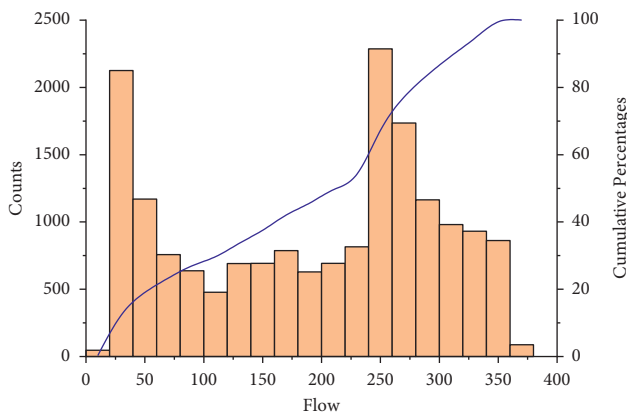


FIGURE 5: Statistical analysis of flow.

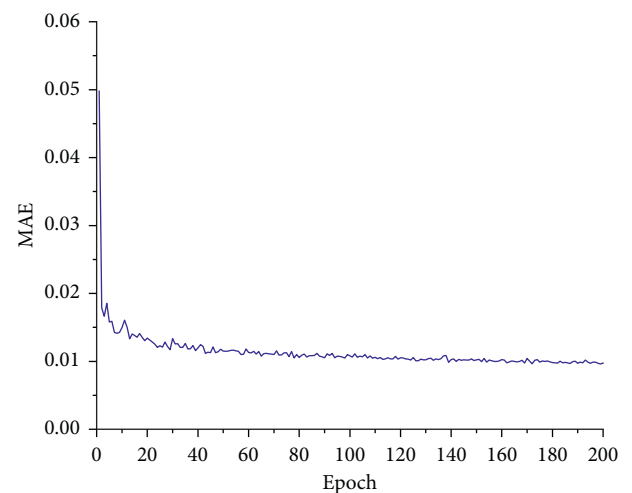


FIGURE 6: Convergence curve of the GA-TCN model.

[filters = 20, kernel size = 3, batch size = 32, and dilations = [1, 2, 4, 8, 16, 32]]. With the increase of the number of iterations, the training error of the GA-TCN model gradually converges, and the mean square error of the hyper-parameter optimal solution decreases to achieve the optimal solution of the search space, as shown in Figure 6.

**4.5. Experimental Results.** Figures 7 and 8 compare the prediction results of the four models on working and weekend days, respectively. All four models can fit the real data well. During the morning peak and evening peak on working days, the flow has the obvious mutation. We choose 6:30 to 9:00 as an example to compare and analyze the prediction model. As can be seen from the enlarged figure of

this period, when the traffic volume changes dramatically, the changing trend of the GA-TCN model is most similar to the actual traffic volume. The prediction error of each model can be obtained by calculating the difference between the predicted value and actual flow. Among them, the average error of LSTM, GRU, TCN, and GA-TCN is 8.40, 5.81, 7.29, and 5.18, respectively. GA-TCN model has the lowest error and the highest prediction accuracy. On weekend days, the peak traffic flow lasts longer and is relatively stable. We choose 13:00 to 18:30 as an example to compare and analyze the prediction model. The predicted results of the GA-TCN model are consistent with the actual traffic flow, while the predicted results of the LSTM and GRU model

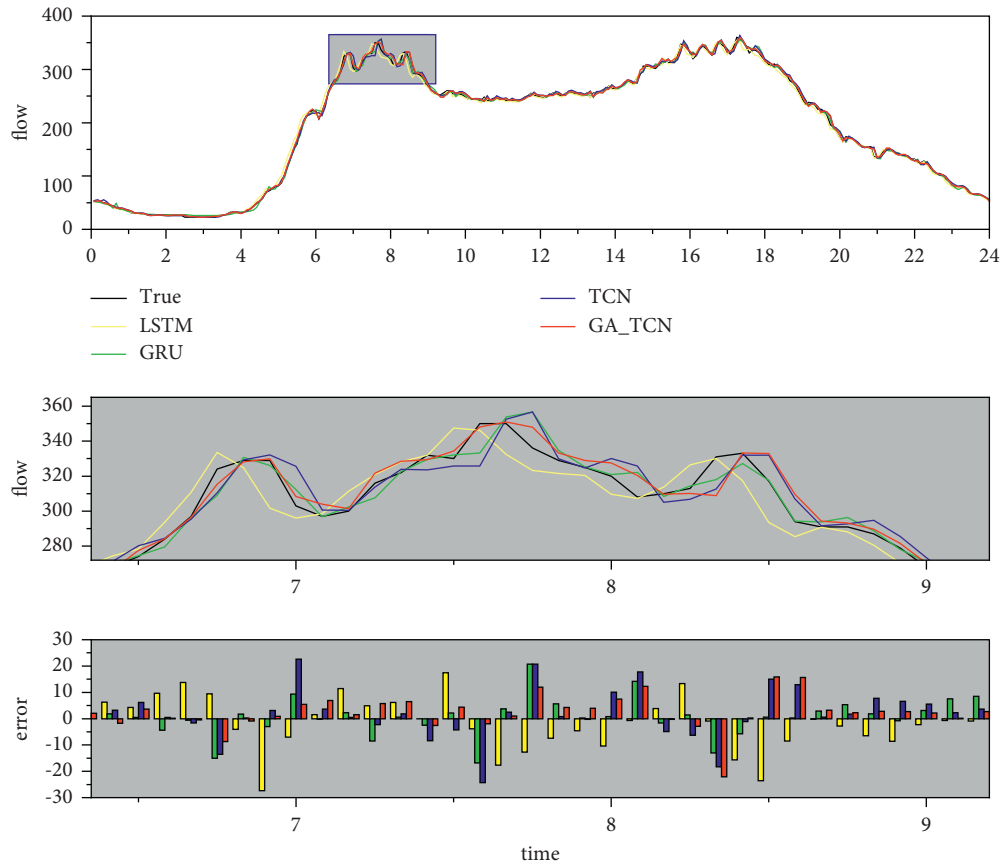


FIGURE 7: Comparison of the real value of traffic flow in a working day.

have an obvious time lag. The average error of LSTM, GRU, TCN, and GA-TCN is 3.82, 4.47, 3.16, and 3.47, respectively. TCN has the smallest mean error, followed by GA-TCN. Although the average error of GA-TCN is larger than that of TCN, the error value of GATCN is basically the same; that is to say, the prediction result of TCN is completely consistent with the change trend of the actual flow. This shows that the GA-TCN model proposed in this paper has strong adaptability and higher prediction accuracy.

Three evaluation indicators, RMSE, MAE, and MAPE, were obtained through the predicted working and weekend days data of four different models, as shown in Table 2. In the evaluation index statistics, the prediction results of the proposed GA-TCN model are the best. The RMSE, MAE, and MAPE of the suboptimal working day model are 18.36%, 28.90%, and 24.69% higher than that of the GA-LSTM model, respectively. The RMSE, MAE, and MAPE of the suboptimal weekend day model are 3.27%, 2.38%, and 5.68% higher than that of the GA-LSTM model, respectively.

**4.6. Analysis of Influencing Factors.** In order to more intuitively show that the proposed GA-TCN model has better accuracy than the other models, Figure 9 shows the RMSE and MAE distribution histograms of the four models according to the data in Table 2. The working day MAE of LSTM, GRU, and TCN is higher than that of the weekend days, while the working day MAE of the GA-TCN is lower than that of the

weekend days. Since the data volume on weekdays was more than twice that of the weekend days, the results demonstrate the GA-TCN proposed in this paper is more suitable for the processing of big data, can more effectively capture the short-term dependence of traffic flow, and improve the accuracy of short-term traffic flow forecasting.

Generally, the time interval for short-term traffic flow forecasting is [5–30] min. For further evaluation, the performances of four models were tested for short-term traffic flow forecasting at different time intervals. The results are shown in Tables 3 and 4.

With the increase of prediction time interval, the prediction difficulty of the model gradually increases. It can be seen from the comparison of prediction performance at different time intervals in Figure 10 that the prediction error of the four models increases roughly linearly. Among them, the change trend and degree of RMSE and MAE of the four working day models are the same. The RMSE, MAE, and MAPE of weekend days LSTM increase by 2.93, 1.91, and 0.5%, respectively, being the most significant changes. The RMSE and the MAE of the GA-TCN increase by 1.34, 1.07, and 0.37%, respectively, being the lowest changes. The results show that the performance of the proposed GA-TCN is better than the existing models at all time intervals. The proposed model has relatively good prediction performance when the data interval is large, and the data sample is small, and it can be better applied to short-term traffic flow forecasting at various time intervals.

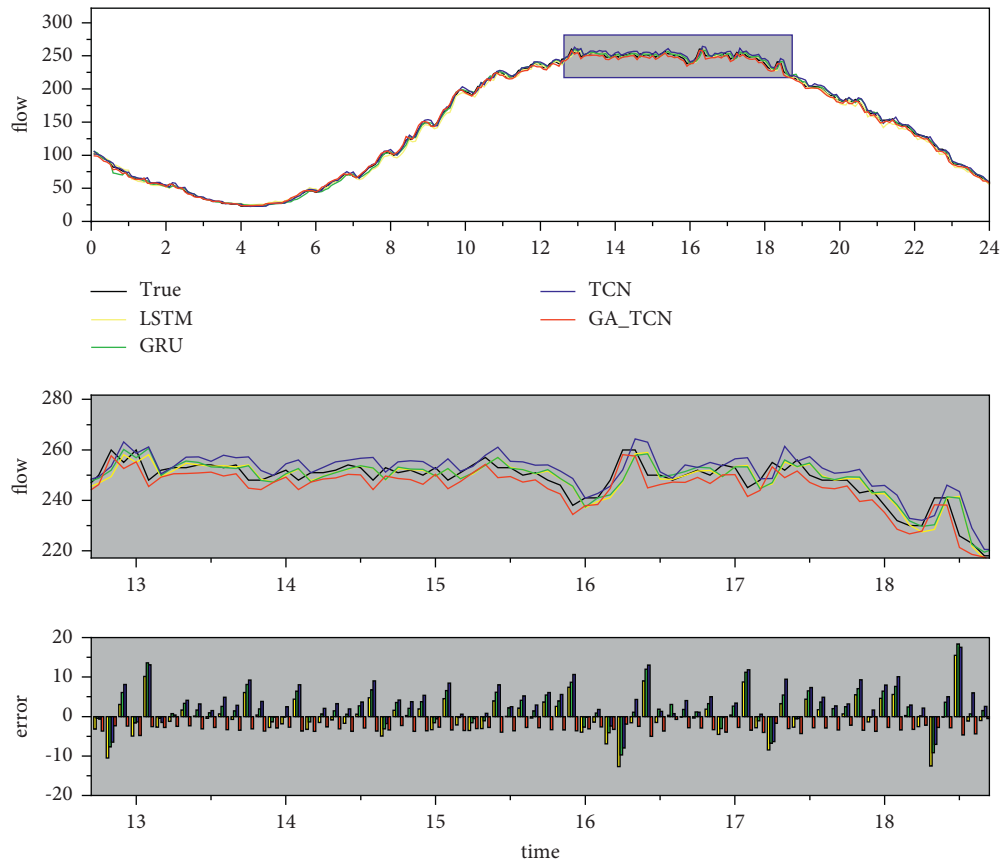


FIGURE 8: Comparison of the real value of traffic flow on a weekend day.

TABLE 2: Comparison of prediction results of four different models.

Model	Workday			Weekend		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
LSTM	5.94	3.99	2.72	5.38	3.89	3.74
GRU	4.77	3.39	2.74	4.11	3.01	2.70
TCN	5.33	3.57	2.02	4.10	3.06	2.42
GA-TCN	4.03	2.63	1.62	3.97	2.94	2.29

#### 4.7. Influence of Parameters on Model Optimization.

Finding the optimal characteristic parameter value by genetic algorithm will reduce the prediction efficiency of the model. In order to reduce the influence of the searching process on the efficiency of prediction, we first selected some samples to preliminarily determine the searching range of TCN characteristic parameters. Then, we make statistics on the optimal individuals of 200 iterations and analyze the relationship between characteristic parameters and error and efficiency, as shown in Figure 11.

It can be seen from Figure 11(a) that MAE increases with the increase of filters, and MAE within the range of 5–15 is relatively low and stable. Figure 11(b) shows that although kernel size has a certain influence on MAE, the influence is small. The impact of batch size on model prediction results is related to the actual data size. You need to adjust the batch size based on the data size. However, under the same number of epochs, a large batch size requires less batch and shorter training time. In other

words, in order to improve the forecasting efficiency, the larger the batch size, the better. However, increasing the batch size will reduce the model generalization performance, and the model performance will decline. This also indicates that in Figure 11(c), when the batch size is 32 or 64, the prediction effect is good, but with the increase of the batch size, the prediction performance gradually decreases. Figure 11(d) shows that with the gradual increase of dilations layer, the model accuracy first improves and reaches the optimal value when dilations layer=4 (dilations=[1, 2, 4, 8]), and then gradually decreases, because the increase of dilations layer leads to a larger receptive field. The benefit of increasing receptive field is to obtain longer temporal dependence of traffic flow. However, with the increase of dilations layer, the number of network layers gradually deepened, and the amount of computation and complexity increased, leading to more difficult training of the model, so the model accuracy decreased.



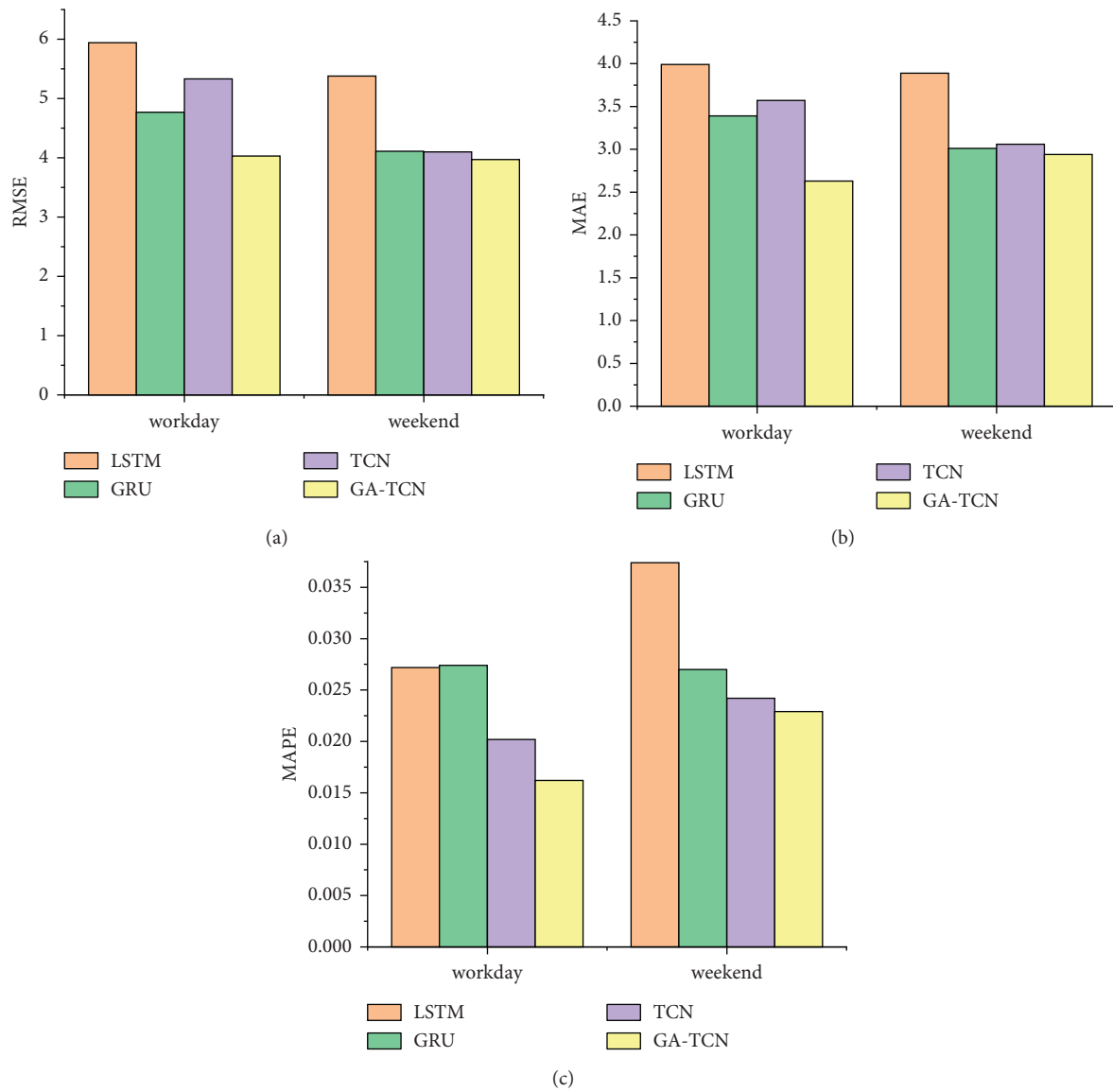


FIGURE 9: Performance comparison of the four models. (a) Comparison of RMSE. (b) Comparison of MAE. (c) Comparison of MAPE.

TABLE 3: Comparison of evaluation indicators on working days.

Model	Evaluation indicators	Time intervals (min)					
		5	10	15	20	25	30
LSTM	RMSE	5.94	6.27	6.48	6.85	7.52	8.08
	MAE	3.99	4.42	4.71	4.97	5.49	5.97
	MAPE	2.72%	2.79%	2.88%	2.92%	2.98%	3.07%
GRU	RMSE	4.77	5.12	5.35	5.89	6.33	7.02
	MAE	3.39	3.89	4.11	4.51	4.92	5.48
	MAPE	2.74%	2.79%	2.85%	2.95%	3.00%	3.09%
TCN	RMSE	5.33	5.53	5.93	6.13	6.73	6.93
	MAE	3.57	3.94	4.28	4.76	5.18	5.51
	MAPE	2.02%	2.11%	2.16%	2.20%	2.28%	2.41%
GA-TCN	RMSE	4.03	4.27	4.55	5.02	5.37	5.79
	MAE	2.63	3.05	3.53	3.81	4.25	4.51
	MAPE	1.62%	1.68%	1.72%	1.82%	1.89%	1.97%

TABLE 4: Comparison of evaluation indicators on weekend days.

Model	Evaluation indicators	Time intervals (min)					
		5	10	15	20	25	30
LSTM	RMSE	4.53	4.92	5.27	5.94	6.76	7.46
	MAE	3.46	3.67	4.21	4.67	4.98	5.37
	MAPE	3.74%	3.83%	3.96%	4.01%	4.10%	4.24%
GRU	RMSE	4.11	4.38	4.79	5.16	5.73	6.12
	MAE	3.01	3.31	3.69	3.91	4.37	4.75
	MAPE	2.70%	2.80%	2.95%	3.16%	3.25%	3.37%
TCN	RMSE	4.1	4.32	4.57	4.82	5.12	5.47
	MAE	3.06	3.22	3.45	3.54	3.91	4.24
	MAPE	2.42%	2.50%	2.56%	2.69%	2.73%	2.85%
GA-TCN	RMSE	3.97	4.21	4.38	4.76	4.93	5.31
	MAE	2.94	3.06	3.32	3.51	3.7	4.01
	MAPE	2.29%	2.34%	2.44%	2.51%	2.58%	2.66%

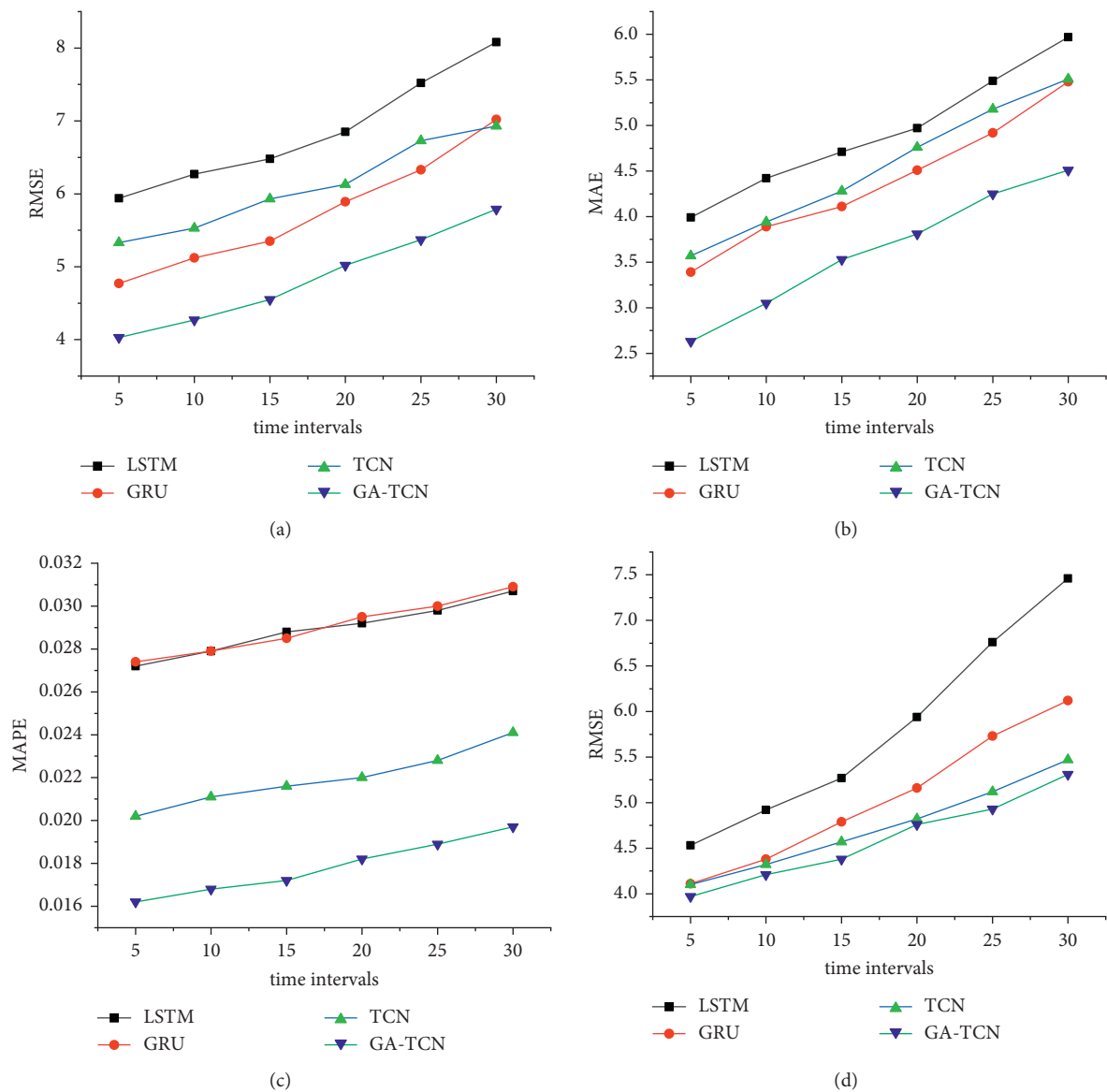


FIGURE 10: Continued.

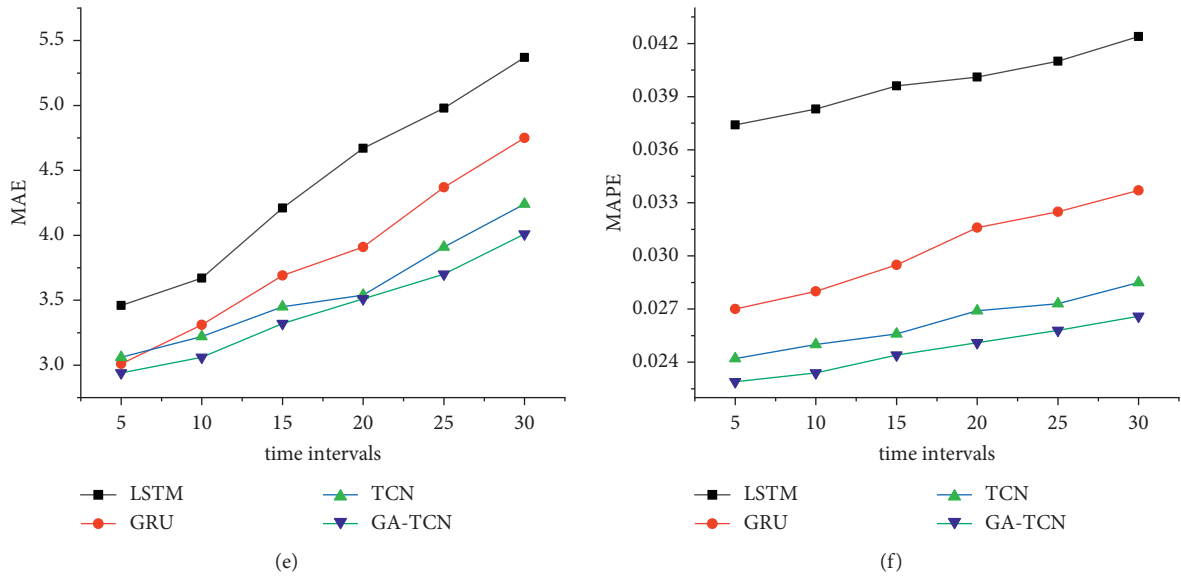


FIGURE 10: Performance comparison of the four models at different time intervals. (a) Comparison of RMSE on working dataset. (b) Comparison of MAE on working dataset. (c) Comparison of MAPE on working dataset. (d) Comparison of RMSE on weekend dataset. (e) Comparison of MAE on weekend dataset. (f) Comparison of MAPE on weekend dataset.

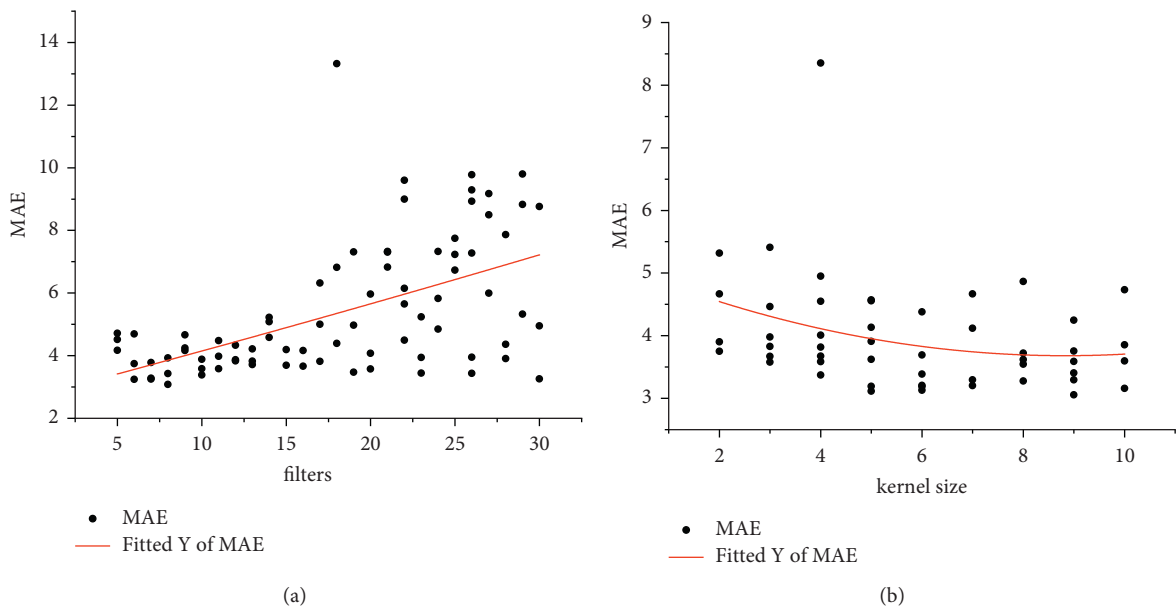


FIGURE 11: Continued.

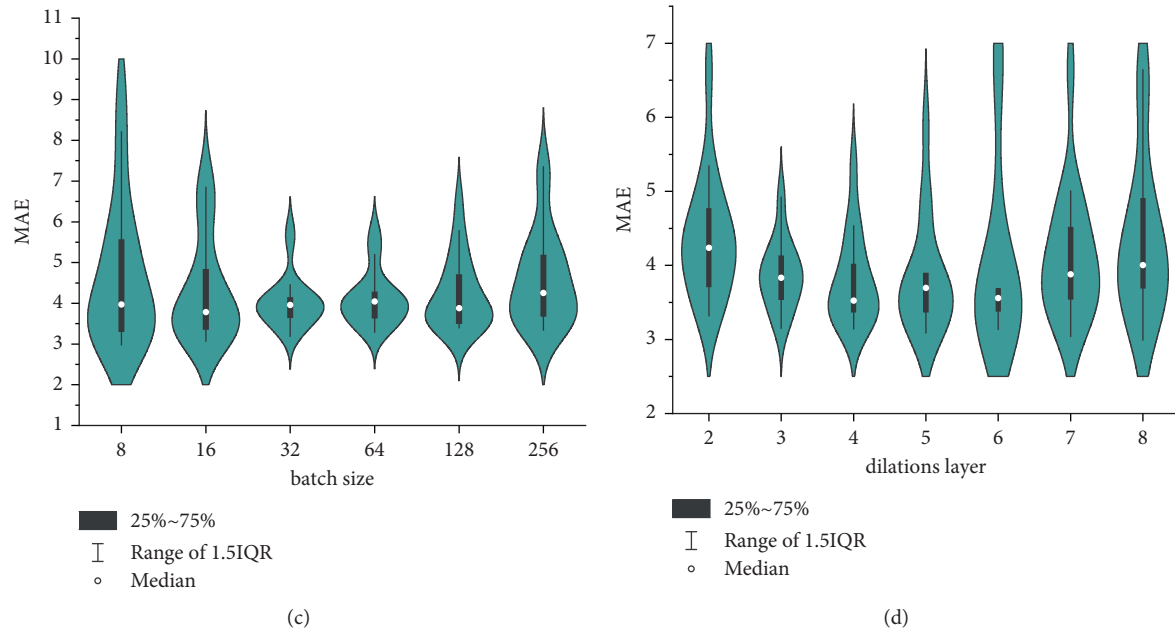


FIGURE 11: The influence of characteristic parameters on MAE. (a) Influence of filters on MAE. (b) Influence of kernel sizes on MAE. (c) Influence of batch size on MAE. (d) Influence of dilations layer on MAE.

## 5. Conclusion

In this paper, a short-term traffic flow forecasting model based on genetic algorithm optimization time convolution neural network is proposed. In the proposed model, the prediction error is taken as the fitness value, and the filters, kernel size, batch size, and dilations of the temporal convolutional neural network are optimized using the genetic algorithm to determine the prediction model with the best accuracy and efficiency. The model is experimentally verified on the PEMS dataset. The experimental results show that the proposed model is better than the existing baseline methods including LSTM, GRU, and TCN and has more adaptability and higher accuracy in short-term traffic flow forecasting.

The experiment conducted to verify the effectiveness of the proposed model was limited to certain parts of the city. In order to establish a more accurate prediction model, the entire urban road network should be used as the research object. In future research, many influencing factors such as traffic accidents, temperature and humidity, and precipitation will be considered.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research was supported in part by the Pacific Northwest 13 Transportation Consortium, US Department of

Transportation University Transportation Center for 14 Federal Region 10, the Science and Technology Planning Project of the Shandong Province (Grant no. 2016GGB01539), and the Science and Technology Planning Project of the Zibo City (Grant no. 2019ZBXC515).

## References

- [1] J. R. Montoya-Torres, S. Moreno, W. J. Guerrero, and G. Mejia, "Big data analytics and intelligent transportation systems," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 216–220, 2021.
- [2] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2019.
- [3] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "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] M. Ahmed and A. Cook, "Analysis of freeway traffic time series data by using box-jenkins techniques," *Transportation Research Record*, vol. 773, no. 722, pp. 1–9, 1979.
- [5] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1678, no. 1, pp. 179–188, 1999.
- [6] 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.
- [7] Y. Qi and S. Ishak, "A hidden Markov model for short term prediction of traffic conditions on freeways," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 95–111, 2014.

- [8] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.
- [9] H. Chang, Y. Lee, B. Yoon, and S. Baek, "Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences," *IET Intelligent Transport Systems*, vol. 6, no. 3, pp. 292–305, 2012.
- [10] W. C. Hong, "Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm," *Neurocomputing*, vol. 74, no. 12, pp. 2096–2107, 2011.
- [11] T.-Y. Hu, "Evaluation framework for dynamic vehicle routing strategies under real-time information," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1774, no. 1, pp. 115–122, 2001.
- [12] R. E. Turochy, "Enhancing short-term traffic forecasting with traffic condition information," *Journal of Transportation Engineering*, vol. 132, no. 6, pp. 469–474, 2006.
- [13] Vlahogianni, Karlaftis, and Golias, "Short-term traffic forecasting: where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, no. 1, pp. 3–19, 2014.
- [14] L. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," *Wiley Inter-disciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 8, p. e1285, 2018.
- [15] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using deep learning theory," *PLoS One*, vol. 10, no. 3, e1285, Article ID e0119044, 2015.
- [16] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, pp. 818–834, 2017.
- [17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modelling*, arXiv, 2014, <https://arxiv.org/abs/1412.3555>.
- [18] M. Coşkun, Ö. Yildirim, A. Uçar, and Y. Demir, "An overview of popular deep learning methods," *Eur. J. Tech*, vol. 7, no. 2, pp. 165–176, 2017.
- [19] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.
- [20] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
- [21] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.
- [22] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*, MIT Press, Cambridge, MA, 2015, <https://arxiv.org/abs/1506.04214>.
- [23] Z. Wu, M. Huang, A. Zhao, and Z. Lan, "Traffic prediction based on GCN-LSTM model," *Journal of Physics: Conference Series*, vol. 1972, no. 1, Article ID 012107, 2021.
- [24] P. Sun, A. Boukerche, and Y. Tao, "SSGRU: a novel hybrid stacked GRU-based traffic volume prediction approach in a road network," *Computer Communications*, vol. 160, pp. 502–511, 2020.
- [25] S. Bai, J. Zico, and K. V. Koltun, *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*, arXiv e-prints, 2018, <https://arxiv.org/abs/1803.01271>.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 90, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [27] D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison-Wesley, New Jersey, 1989.