

Research Article

Mobile Virtual Reality Rail Traffic Congestion Prediction Algorithm Based on Convolutional Neural Network

Yaxiang Li  and Yuanqing Wang 

College of Transportation Engineering, Chang'an University, Xi'an, Shaanxi 710064, China

Correspondence should be addressed to Yaxiang Li; 201812210203028@zcmu.edu.cn

Received 5 May 2022; Accepted 8 June 2022; Published 24 June 2022

Academic Editor: Muhammad Muzammal

Copyright © 2022 Yaxiang Li and Yuanqing Wang. 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.

In order to explore a mobile virtual reality railway traffic congestion prediction algorithm based on convolutional neural network, an expanded causal convolution neural network (DCFCN) was proposed, which introduced the expanded convolution to increase the size of the receptive field and obtain the long-term memory of the sequence. At the same time, causal convolution is introduced to solve the problem of information leakage. DCFCN is made up of 6 convolutional layers, each layer achieves causal convolution through padding, and the expansion coefficient increases exponentially layer by layer. Experimental results show that LSTM and GRU can obtain the time sequence relationship of mobile virtual reality traffic flow sequence, and the prediction effect is better than simple method and traditional ARIMA model, but still inferior to DCFCN. The RMSE of DCFCN decreased by 0.38 compared with single-layer LSTM, 0.52 compared with double-layer LSTM, and 0.38 compared with single-layer and double-layer GRU. It shows that TCN model can indeed do better than RNN in sequence modeling. It is proved that the proposed DCFCN is superior to other comparison models in mobile virtual reality traffic flow prediction, and the computational efficiency on GPU is significantly improved.

1. Introduction

Mobile virtual reality (VR) technology is an interactive simulation of virtual environment generated by computer, which makes users feel like they are on the scene through visual, auditory, and tactile effects. It is a kind of natural human-machine interface between human and the virtual environment generated by computer. Virtual reality platform consists of computer, graphics acceleration card, high-frequency display, helmet display (or stereoscopic glasses), tracking sensor, data glove, 3d mouse, and other hardware. In recent years, with the promotion of urbanization and the rapid increase in the number of motor vehicles, traffic congestion has become increasingly serious and become an important issue affecting urban development [1]. In order to effectively solve the problem of traffic congestion, some scholars propose to use the method of traffic flow prediction to solve the problem [2]. The traffic flow prediction method predicts the traffic flow information for a period of time in

the future through the established traffic flow prediction model, so as to provide reference information for traffic management and travel of traffic managers and travelers, so as to avoid traffic congestion [3]. BP neural network is a multilayer feed-forward neural network, which has become an important method for research on traffic flow prediction due to its good self-learning ability, generalization ability, and nonlinear mapping ability [4]. Due to the randomness of the initial weights and thresholds selected by BP neural network, it has poor global search ability and is easy to fall into the local optimal solution and slow convergence rate [5].

In order to overcome the shortcomings of BP neural network, in traffic flow prediction, some scholars use Particle Swarm Optimization (PSO) algorithm and Cuckoo Search (CS) algorithm to optimize the initial weight and threshold of BP neural network. For example, Qian et al. proposed the improved BP neural network IPSO-BP, which gives a PSO algorithm IPSO of adaptive mutation operator, and is

applied to optimize the initial weight and threshold of BP neural network [6]. Ma et al. proposed the improved BP neural network CPSO-BP, which used the chaotic PSO algorithm CPSO to optimize the initial weight and threshold of the BP neural network [7]. Zhao et al. proposed the improved BP neural network GCS-BP, which used the CS algorithm GCS based on Gaussian disturbance to optimize the initial weights and thresholds of the BP neural network [8]. Wang et al. proposed the improved BP neural network TCS-BP, which used the T-distributed CS algorithm TCS to optimize the initial weight and threshold of BP neural network [9]. However, the existing optimization algorithm for the initial weight and threshold value of BP neural network optimization results are not ideal. Wang et al. built a simulator similar to the actual cab and a virtual world with realistic 3D scenes, and the instrument display, sound, visual scene, site, weather effect, and time effect were all the same or similar to the actual and simulate kinematic and dynamic behavior while driving on the ground. Through helmet, tracker, control lever, and other equipment, users can immerse themselves in and interact with it and complete driving, storage, loading and unloading tasks, as well as vehicle collision and accident simulation. The immersive mobile virtual reality system is as realistic and real time as the external world seen through a real window [10].

In view of this research problem, this paper proposes therefore traffic flow prediction is the key to realize traffic control and traffic guidance in intelligent transportation system. Traditional one-dimensional convolutional neural network (CNN) is difficult to obtain long-term memory in traffic flow prediction and has the problem of information leakage. In this paper, extended causal convolutional neural network (DCFCN) is proposed. Extended convolution is introduced to increase the receptive field size and obtain the long-term memory of the sequence. Meanwhile, causal convolution is introduced to solve the problem of information leakage and construct 3D road traffic model and basic information platform. The virtual road environment can provide a variety of services, dynamic, fast, high precision, standardized access and storage of road traffic spatial and attribute information and fast and easy to enjoy information query, retrieval, and statistics. Experimental results show that LSTM and GRU can obtain the time sequence relationship of mobile virtual reality traffic flow sequence, and the prediction effect is better than simple method and traditional ARIMA model, but still inferior to DCFCN. The RMSE of DCFCN decreased by 0.38 compared with single-layer LSTM, 0.52 compared with double-layer LSTM, and 0.38 compared with single-layer and double-layer GRU. It is proved that the proposed DCFCN is better than other comparison models in traffic flow prediction, and the computational efficiency on GPU is significantly improved.

2. Network Structure

2.1. Virtual Environment for Road Planning. Mobile virtual reality and geography information system is used. In this system, modern high technologies such as the urban

geographical environment, infrastructure, natural resources, ecological environment, population distribution, and human landscape are utilized. All kinds of information in digital collection and storage, dynamic monitoring and processing technique, and comprehensive management are all used to build the three-dimensional road traffic model and the basic information platform. Through the virtual road environment can provide a variety of services, dynamic, fast, high precision, standardized access, and storage of road traffic spatial and attribute information and fast and easy to enjoy information query, retrieval, and statistics.

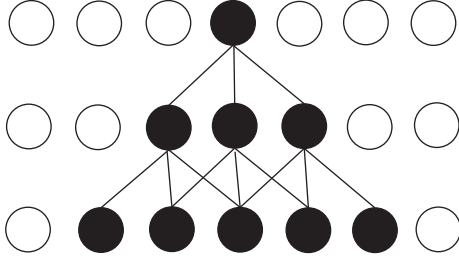
By applying mobile virtual reality technology to urban traffic, designers can directly plan and design roads in 3d virtual city. By intuitively understanding the environment around the road, the coordination relationship between the road and the environment and the impact on traffic safety can be considered more comprehensively. In addition, designers can also immerse themselves in virtual city traffic through corresponding sensory devices, driving virtual cars to experience the capacity of virtual city traffic and the management and control of each traffic intersection, so as to better optimize the road design.

2.2. Time Convolutional Neural Networks

2.2.1. Extended Convolution. In the convolutional neural network, receptive field refers to the size of the mapping region of the nodes on the FeatureMap on the input image. In order to better acquire long-term memory, the key is to enlarge the receptive field [11]. As shown in Figure 1, dark colors represent the area “seen” by a node at layer 3. When the number of convolution layers is L and the size of each convolution kernel is K , the receptive field size is $(k-1) \times L + 1$. The receptive field size of the convolutional neural network is linearly related to the size of the convolutional kernel and the number of convolutional layers [12]. Therefore, increasing the convolution layer and the convolution kernel can enlarge the receptive field. However, the deeper number of convolutional layers and larger convolutional kernels make the number of network parameters huge and difficult to complete the training [13, 14]. Three 3×3 convolution kernels are stacked to make their receptive fields have the same size as that of a 7×7 convolution kernel. However, the number of parameters of three 3×3 convolution kernels is about $27C$ (C represents a constant), and the number of parameters of a 7×7 convolution kernels is $49C$. In the convolutional neural network, it is usually preferred to use the convolution kernel that is not too large. A large convolution kernel will dramatically increase the network parameters and increase the operational complexity [15]. In addition, increasing the stride or pooling can also better obtain the long-term memory, but may cause serious information loss.

Another method is to use extended convolution. Compared with general convolution, in addition to the size of the convolution kernel, the dilation rate used to represent the size of the extended convolution is increased [16].

The calculation formula of extended convolution is defined as

FIGURE 1: Receptive field ($k=3, L=2$).

$$F(s) = (x^*_d f)(s) = \sum_{i=0}^{k-1} f(i)x_{s-d-i}, \quad (1)$$

where d represents the expansion coefficient and k represents the size of the convolution kernel. When D is 1, the extended convolution degenerates into ordinary convolution. By controlling the size of D , the receptive field can be widened under the premise of constant calculation amount [17].

2.2.2. Causal Convolution. Extended convolution solves the problem of “long-term memory” well, while causal convolution solves the problem of information leakage. Information leakage is a problem of sequence processing such as traffic flow prediction by indicators. It is necessary to ensure that the model cannot reverse the sequence order. When the model predicts time T , data of future time such as $T+1$ and $T+2$ cannot be used [18]. However, in the traditional one-dimensional convolutional neural network, the convolution checks the time data before and after the convolution calculation, which inevitably uses the future time data for modeling, that is, information leakage. Causal convolution was first proposed in Wave Net. In the causal convolution, the output at time T only convolved with the input at time T and earlier in the previous layer and had nothing to do with the value after time T [19]. Compared with the traditional convolutional neural network, the causal convolution can only “see” the data in the past, but cannot “see” the data in the future, so the information leakage is well solved. One-dimensional causal convolution is generally realized by padding. The front end of the sequence is filled with zeros of corresponding bits, while the end of the sequence is not filled [20].

2.3. Network Structure. Based on the basic idea of TCN, the above extended convolution is combined with causal convolution, and TCN is first applied in the field of traffic flow prediction. Figure 2 for the proposed network structure, 1 for traffic flow sequence of input layer, the next six stacked layer 1d convolution, each layer convolution is achieved by padding causal convolution, convolution kernels is 4 size, number of convolution kernels is 32, each layer of expansion coefficient, respectively, 1, 2, 4, 8, 16, 32, then access the connection layer, output the traffic flow of the next moment. The specific steps are as follows:

Step 1: first, the traffic flow sequence data are processed in the way of sliding window. The window size is 21, and one data is slid forward each time, that is, the time step is 21, to predict the traffic flow size at the next moment.

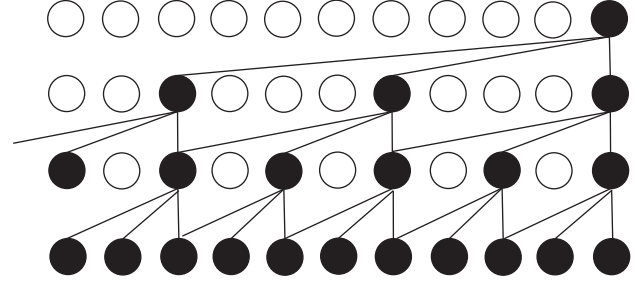


FIGURE 2: Multilayer extension-causal convolution stack.

Step 2: preprocess the data after sliding window processing and divide the training set and test set.

Step 3: input the training set into the DCFCN model for training.

Step 4: use the trained model to predict the traffic flow size of the test set and calculate the error.

Figure 2 shows the schematic diagram of multilayer extended causal convolution stack, and the DCFCN expansion coefficients are 1, 2, and 4, respectively. The expansion coefficient increases exponentially, so does the size of receptive field. Even in the case of a few convolution layers, a very large receptive field can be obtained, and the computational efficiency of the model is also guaranteed.

The convolutional neural network computes multiple convolution in parallel in the convolution level and generates a set of linear excitation responses, so it can be accelerated using gpus. Each linear activation response goes through a nonlinear activation function, such as the linear rectified activation function (ReLU). Layer regularization is performed prior to ReLU activation, limiting the output of the layer to an interval of 0 to 1.

2.4. Plane Normal Model. Orbital plane design also requires the design of edge lines, which requires unified calculation and the establishment of normal model. The so-called normal model is a method to calculate points based on the center line along the normal direction. On unbroadened and fully broadened tracks, a sideline is a curve with equal distance from the center line, which is mathematically called an isometric curve. The calculation formula of constant distance curve is as follows:

$$\begin{aligned} x_t &= x - B \times \sin(\alpha), \\ y_t &= y + B \times \cos(\alpha), \end{aligned} \quad (2)$$

where x_t, y_t are the coordinates of road sideline; x, y is the coordinate of road center line; B is side road width; and α direction Angle.

The above is the so-called normal model. Although the model is simple, it plays a very important role in road route design. The plane coordinates of all points deviating from the center line by a certain distance can be calculated by the above formula. The normal model can be used to calculate the coordinates of points at any distance from the center line, which is the basis for drawing each lane line of the road. The

plane coordinates of points on each side line corresponding to each demand point on the center line can be calculated in turn, and the road side line can be drawn by connecting these coordinate points.

Intersection is the key of urban traffic and the bottleneck area of the whole urban road network, so the establishment of intersection model is particularly important in the modeling of urban rail. An intersection is formed when two sections of road or two roads meet.

2.5. Spatial Measurement Based on Mobile Virtual Reality

- (1) Spatial distance is calculated on the basis of spatial coordinate query. The space query realizes the transformation from window coordinates to world coordinates, and then the distance between two points can be calculated in the three-dimensional space coordinate system according to the theory of space geometry, and the total distance can be obtained by summation.

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}. \quad (3)$$

- (2) The specific steps of calculation are as follows: whenever the user adds the measured key points by clicking the mouse in the scene, the 3d coordinates of the picked points are stored in a dynamic array. When the measurement is finished by right clicking, the key points are taken out from the dynamic array one by one, according to the distance formula in 3d coordinates:

$$S(A, B, C) = \begin{vmatrix} X1, X2, X3 \\ Y1, Y2, Y3 \\ 1, 1, 1 \end{vmatrix} * 0.5. \quad (4)$$

3. Experiment

3.1. Interactive Scene Design. In order to ensure the real-time performance of models in 3d scenes and make them truly reflect the latest information, models are usually added or deleted in the scene according to the actual situation. In general, models are added to virtual scenes through the Lynx Prime graphical interface, which is convenient and simple. However, if you want to add and remove models in real time while the scenario is running, this approach is obviously not sufficient. Dynamic add and delete model is convenient for users to manage the scene, and users can modify the scene object in real time according to their own needs. By adding or deleting models or scaling, moving, and rotating models to better meet the needs of the scene, this greatly improves the situation that users can only browse the scene and operate the model simply. Users can not only roam the scene but also organize new scenes by modifying the model, which is more in line with humanized design. It can add models, delete selected models, delete multiple models within the

TABLE 1: Experimental environment.

	Project	Parameter
CPU machine	CPU	Intel(R) Xeon(R) CPU E5-2670
	Memory	64G
CPU machine	CPU	Intel(R) Xeon(R) CPU E5-2603
	CPU	NVIDIA Ge Force GTX 1080 Ti
	Memory	64G
	Memory	11G
Version	Lioux	Ubuntu 14.04.5 LTS
	Python	3.5.4
	TensorFlow	1.4.0
	Keras	2.1.2

scope, and modify model attributes in real time, such as zoom and position movement.

3.2. Experimental Environment. In order to compare the performance of different models on CPU and GPU, experiments will be conducted on CPU and GPU, respectively, in this paper. The experimental environment is shown in Table 1.

3.3. Data and Preprocessing. The experimental data came from the PEMS system. Eight detectors at different locations were randomly selected and numbered VDS1213963, VDS1201453, VDS1201637, VDS1201671, VDS1201705, VDS1201735, and VDS1201751. The traffic flow data of each detector from April 1, 2019 to June 30, 2020, were obtained at a time interval of 5 min. Figure 3 shows the traffic flow changes of VDS1213963 detector during the week from June 22, 2019 to June 28, 2020.

Due to equipment failure, noise interference, improper storage, human error, and other emergencies, the data cannot be collected 100%, and there are error data and data loss. Therefore, the original data must be filled before the prediction model is established. There are three methods of data filling: prediction, interpolation, and statistical learning. The prediction method is to build the prediction model according to the historical data and predict the lost data. The interpolation method uses the historical data or the data of adjacent detectors directly to fill in the lost data. Statistical learning method is to fill according to statistical characteristics. Usually, the probability distribution model of data is assumed first, and then the parameters of the model are estimated iteratively.

Interpolation is used, using historical averages of traffic flows to fill in. Considering the different traffic flow patterns in different days of a week, the traffic flow data were divided into 7 categories from Monday to Sunday, and the average traffic flow at all times of a day in each category was calculated. The average traffic flow was used to fill in the missing values. Fill in the missing data of detector no. VDS1213963 on March 8, 2019.

The minimum and maximum normalization is adopted to map the data to the interval [0, 1]. The normalization formula is shown in formula (5):

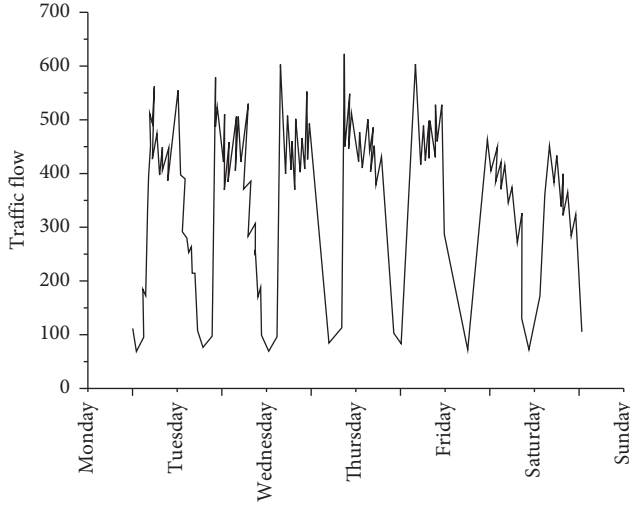


FIGURE 3: Traffic flow in a week.

$$z = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (5)$$

where z represents the normalized data, x represents the original data, x_{\min} is the minimum value of x , and x_{\max} is the maximum value of x .

3.4. Evaluation Indicators. In order to evaluate the quality of the traffic flow prediction model, the root mean square error (RMSE) is used as the main evaluation index, supplemented by observing the change of mean absolute error (MAE) and mean absolute percentage error (MAPE). When MAPE is calculated, the zero value sample is eliminated.

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|^2}, \\ MAE &= \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \\ MAPE &= \frac{100}{n} \sum_{i=1}^n |(\hat{y}_i - y_i)/y_i|. \end{aligned} \quad (6)$$

3.5. Experimental Settings. The models used in the experiment in this section include random walk, historical mean, ARIMA, LSTM, GRU, and DCFCN. Since different hyperparameter settings will affect the prediction accuracy of the model, the optimal parameters of different models are obtained by consulting relevant literature or repeated cross validation on the premise of ensuring fairness as far as possible. Among them, the time step of LSTM, GRU, and DCFCN is 21, and the data time interval is 5 min.

The parameters are set as follows:

- (1) For ARIMA, $p=2$, $d=1$, and $q=1$ are taken, respectively. In order to obtain a stationary sequence, a first-order difference is made in advance.

Considering that the traffic flow has a weekly periodicity, the traffic flow at the current moment is similar to that at the same time last week, and the difference is used to subtract the traffic flow at the same time last week from the traffic flow at the current moment.

- (2) LSTM1 and LSTM2 represent single-layer LSTM and double-layer LSTM, respectively, and the number of hidden layer units is 64;
- (3) GRU1 and GRU2 indicate that single-layer GRU and double-layer GRU are used, respectively, and the number of hidden layer units is 64;
- (4) The expansion coefficient of each layer of DCFCN is [1, 2, 4, 8, 16, 32], the number of convolutional kernels of each layer is 32, and the size of convolutional kernels is 4.

4. Experimental Results and Analysis

4.1. Experimental Results. The experimental results of traffic flow prediction for a single detector are shown in Table 2, where RMSE, MAE, and MAPE are all averaged after 10 experiments, and S(RMSE) represents the sample standard deviation of RMSE.

As can be seen from Table 2, compared with other comparison models, the proposed DCFCN has the best prediction effect and the lowest in RMSE, MAE, and MAPE indicators. LSTM and GRU can obtain the temporal relationship of traffic flow sequence, and their prediction effect is better than simple methods and traditional ARIMA model, but still inferior to DCFCN. Compared with single-layer LSTM, DCFCN on RMSE is reduced by 0.38, 0.52, and 0.38, respectively, compared with double-layer LSTM, and compared with both single-layer and double-layer GRU. It shows that TCN model can do better than RNN in sequence modeling. In order to verify whether the DCFCN model is valid on different detectors, different detectors are selected for repeated tests. Although the prediction effect of two detectors is not as good as that of LSTM or GRU, DCFCN has faster training speed than other models while ensuring the prediction effect. The output of LSTM, GRU, and other recurrent neural networks at each moment is generated by performing the same operation on the output at the previous moment. This inherent property enables RNN to have “long-term memory.” It is this feature that makes RNN computable only serialized, not parallelized or accelerated by GPU. While TCN is based on convolutional neural network, there is no need to update and retain “memory,” and there is no dependency between outputs, so it is easy to parallelization and can be trained more efficiently on GPU.

Table 3 shows the training speed of different models. CPU time and GPU time, respectively, represent the time required for a round of iterative training of the model. As can be seen from the table, LSTM and GRU did not gain any speed on the GPU machine and even decreased. The reason is that the CPU hardware resource of the EXPERIMENTAL

TABLE 2: Experimental results.

Model	RMSE	MAE	MAPE/%	S (RMSE)
RW	28.6168	20.8329	10.3694	—
HA	26.0555	18.7785	8.8641	—
ARIMA	29.4193	21.6385	10.7134	—
LSTM1	23.5795	17.3004	8.5775	0.0921
LSTM2	23.7238	17.4530	8.7664	0.0788
CRU1	23.5875	17.3313	8.7126	0.0916
CRU2	23.5834	17.3427	8.6599	0.0930
DCFCN	23.1993	16.9903	8.4129	0.0868

TABLE 3: Training speed.

Model	RMSE	Time/s	
		CPU	CPU
LSTM1	23.5795	10.65	15.51
LSTM2	23.7238	21.91	31.08
GRU1	23.5875	11.59	12.08
GRU2	23.5834	24.34	24.74
DCFCN	23.1993	28.33	4.17

GPU machine is not as good as that of the CPU machine. The DCFCN model has obvious acceleration effect on the GPU machine.

4.2. Hyperparameter Analysis. Norm relu means that the output is first layerized and then activated by the relu activation function. Wavenet represents the activation function used in the Wavenet model, which is essentially the bitwise multiplication of neurons activated by Tanh function and Sigmoid function, respectively. The Norm_relu activation function used in the end works best. In addition, with the gradual increase of the expansion coefficient, the accuracy of the model first improves and then decreases. The reason is that the increase of the expansion coefficient leads to the increase of the receptive field, and the benefit brought by the increase of the receptive field is that the temporal dependence of traffic flow can be obtained for a longer time. However, with the increase of the expansion coefficient, the network layer is gradually deepened, and the calculation amount and complexity increase, which makes the model more difficult to train, so the accuracy of the model decreases. When training deep TCN model, residual network should be used to assist training. However, the addition of residual network will inevitably make the model more difficult to train, so the accuracy of the model will decrease. When training deep TCN models, residual networks should be used to aid training. However, the addition of residual network will inevitably increase the training time of the model and make it more difficult to find the optimal hyperparameters. The introduction of Dropout in DNN can effectively prevent the phenomenon of overfitting, but the removal of the Dropout mode of neural nodes in DCFCN will reduce the accuracy of the model. Finally, when the DCFCN activation function is Normrelu, the convolution kernel size is 4, the number of convolution kernels is 16, and the expansion coefficient is 32, and when the Dropout

mechanism is not introduced, the prediction results are the best.

5. Conclusion

Based on the basic idea of time convolutional neural network, a DCFCN model is proposed and applied to mobile virtual reality traffic flow prediction. DCFCN consists of six convolution layers stacked, each layer realizes the causal convolution by means of padding, and the expansion coefficient increases exponentially by layer. The exponential expansion coefficient enables the model to obtain the long-term memory of the sequence. Experimental results show that the prediction effect of DCFCN is better than LSTM, GRU and other recurrent neural networks, which proves the feasibility of time convolutional neural network in traffic flow prediction. At the same time, because convolutional neural network is naturally easy to parallel, the training efficiency of DCFCN on GPU has been significantly improved.

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 they have no conflicts of interest.

References

- [1] R. Zhang, M. Liu, Y. Yin, Q. Zhang, and Z. Cai, "Prediction algorithm for network security situation based on bp neural network optimized by sa-soa," *International Journal of Performance Engineering*, vol. 16, no. 8, p. 1171, 2020.
- [2] S. G. Meshram, M. A. Ghorbani, S. Shamshirband, V. Karimi, and C. Meshram, "River flow prediction using hybrid psogsa algorithm based on feed-forward neural network," *Soft Computing*, vol. 23, no. 20, pp. 10429–10438, 2019.
- [3] X. Meng, L. Nie, and J. Song, "Iot individual privacy features analysis based on convolutional neural network," *Cognitive Systems Research*, vol. 57, pp. 126–130, 2019.
- [4] Z. Wang, N. Li, T. Wu, H. Zhang, and T. Feng, "Simulation of human ear recognition sound direction based on convolutional neural network," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 209–223, 2020.
- [5] Z. Huang, G. Huang, Z. Chen, C. Wu, X. Ma, and H. Wang, "Multi-regional online car-hailing order quantity forecasting based on the convolutional neural network," *Information*, vol. 10, no. 6, p. 193, 2019.
- [6] Y. S. Qian, J. W. Zeng, S. F. Zhang, D. J. Xu, and X. T. Wei, "Short-term traffic prediction based on genetic algorithm improved neural network," *Tehnički Vjesnik*, vol. 27, no. 4, pp. 1270–1276, 2020.
- [7] Y. Ma, L. Li, Z. Yin, A. Chai, M. Li, and Z. Bi, "Research and application of network status prediction based on bp neural network for intelligent production line," *Procedia Computer Science*, vol. 183, no. 20, pp. 189–196, 2021.
- [8] J. Zhao, H. Tian, and D. A. Li, "Channel prediction based on bp neural network for backscatter communication networks," *Sensors*, vol. 20, no. 1, p. 300, 2020.

- [9] S. Wang, Z. Fan, Z. Li, H. Zhang, and C. Wei, "An effective lunar crater recognition algorithm based on convolutional neural network," *Remote Sensing*, vol. 12, no. 17, p. 2694, 2020.
- [10] W. Wang, L. Wu, Y. Huang, H. Wang, and R. Zhu, "Link prediction based on deep convolutional neural network," *Information*, vol. 10, no. 5, p. 172, 2019.
- [11] Y. Xiao and Q. Nian, "Vehicle location prediction based on spatiotemporal feature transformation and hybrid lstm neural network," *Information*, vol. 11, no. 2, p. 84, 2020.
- [12] Y. Zhang, J. Zhang, L. Luo, and X. Gao, "Optimization of lmbp high-speed railway wheel size prediction algorithm based on improved adaptive differential evolution algorithm," *International Journal of Distributed Sensor Networks*, vol. 15, no. 10, 2019.
- [13] Z. Jiang, J. Jiang, Q. H. Yao, and G. Yang, "A fast solver based on deep neural network for difference equation," *Lixue Xuebao/Chinese Journal of Theoretical and Applied Mechanics*, vol. 53, no. 7, pp. 1912–1921, 2021.
- [14] X. Xu, N. Hu, T. Li et al., "Distributed temporal link prediction algorithm based on label propagation," *Future Generation Computer Systems*, vol. 93, pp. 627–636, 2019.
- [15] W. Jianguo, L. Zhe, L. Yandong, D. Lijuan, and L. Zhiwei, "Fouling resistance prediction based on ga-elman neural network for circulating cooling water with electromagnetic anti-fouling treatment," *Journal of the Energy Institute*, vol. 92, no. 5, pp. 1519–1526, 2019.
- [16] D. Yu, J. Guo, Q. Zhao, and J. Hong, "Prediction of the dynamic performance for the deployable mechanism in assembly based on optimized neural network," *Procedia CIRP*, vol. 97, no. 3, pp. 348–353, 2021.
- [17] L. Wan, H. Li, Y. Chen, C. Li, and E. Sciubba, "Rolling bearing fault prediction method based on QPSO-BP neural network and dempster-shafer evidence theory," *Energies*, vol. 13, no. 5, p. 1094, 2020.
- [18] M. Gu, Y. Wei, H. Pan, and Y. Ying, "A new real-time pinch detection algorithm based on model reference kalman prediction and srms for electric adjustable desk," *Sensors*, vol. 20, no. 17, p. 4699, 2020.
- [19] M. Mehrabi, M. Mohammadkarimi, M. Ardakani, and Y. Jing, "Decision directed channel estimation based on deep neural network k -step predictor for MIMO communications in 5G," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2443–2456, 2019.
- [20] M. N. Meqdad, R. A. Akam, and S. Kadry, "New prediction method for data spreading in social networks based on machine learning algorithm," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 6, pp. 3331–3338, 2020.