Traffic Flow Prediction by One-dimensional Convolutional Neural Network

Yifu Deng
School of Automobile, Chang'an University
Xi'an, China
dengyifu99@gmail.com

Abstract—**Traffic congestion is a problem that cannot be ignored in social development. To improve the efficiency of traffic travel, accurate traffic flow prediction is necessary. This paper uses a one-dimensional convolutional neural network to predict the traffic flow, processes the data, and compares it with LR, SVM, Bayes, and Decision Tree methods. The results obtained by using RMSE, MAE, and R^2 as evaluation indicators are more reliable than other methods, and the results are illustrated through data visualization. Finally, the advantages of deep learning and the future development direction are introduced.**

Keywords—Deep Learning, One-Dimensional Convolutional Neural Network, Traffic Flow Prediction, SVM, Bayes

I. INTRODUCTION

Recently, with the increase in the number of people and the improvement of people's living standards, the number of cars has also increased, and the demand for people's travel has also increased. With this comes the increase in the probability of traffic congestion and traffic accidents. Therefore, Accurate prediction of traffic flow has also become crucial [1]. As a vital part of the intelligent transportation system, traffic flow prediction is of great significance to the improvement of traffic congestion and the protection of traffic safety [2], and it also has a far-reaching impact on the happiness of people's lives, the good operation of the city, and the development of the economy.

Traffic flow is closely related to many factors, such as weather, holidays, and other conditions that significantly impact road traffic flow. For example, the traffic flow in rainy and snowy weather is less than that on sunny days, and the amount of people traveling on holidays is larger. However, it is inaccurate to predict traffic flow only through a single traffic feature without considering other factors, greatly increasing the uncertainty of the prediction results [3]. Therefore, we need to consider other conditions like weather and holidays when studying traffic flow to make a more accurate prediction. In this paper, we proposed to use a convolutional neural network (CNN) to predict traffic flow. This algorithm has the characteristics of fast and convenient data processing, and at the same time, it has a high degree of automation. It can also train the model within a certain period of time [4], applicable to the traffic flow forecast in this article, considering the weather and holidays.

The structure of this paper can be divided into the following parts: In section Ⅱ, we introduce the related work, mostly focusing on the machine learning application of traffic flow prediction; In section Ⅲ, We preprocessed the data, and got the holiday data in the United States, and finally got the feature description table; In section Ⅳ, we introduce the characteristics, structure, and formula of CNN, and give an example to illustrate; In section Ⅴ, We introduced the variables we used and the advantages and disadvantages of each variable; In section Ⅵ, we get better results for 1DCNN and use tables and box plots to help illustrate; In section Ⅶ, we introduce the advantages of deep learning and the improvement direction in the future.

II. RELATED WORK

In this section, we introduced some related work regarding some machine learning applications in traffic flow prediction. In 2020, Andrew Moses proposed a series of machine learning methods to predict traffic volume, and they used a dataset consisting of daily volumes of traffic across various stations in the US and achieved much time on making traffic predictable [5]. In addition, Sun introduced a novel wavelet-SVM to predict short-time passenger flow, their model includes three important parts which are decomposition, prediction stage, and reconstruction, which achieved the best forecasting performance compared with the state-of-the-art methods [6]. Xinglong Luo proposed to predict Spatiotemporal Traffic Flow based on the combination of k-nearest neighbor (KNN) and long short-term memory network (LSTM), and they conduct the result-level fusion with rank-exponent weighting method to obtain the final prediction results and yield a promising prediction performance [7].

On the other hand, Shiliang Sun came up with a method to forecast the traffic flow based on Bayesian networks, and they modeled it as a Bayesian network by linking the traffic flows among neighbor roads in the same traffic network, he proposed that this method was different from others because it involved the information of adjacent road, and it also included the problem of traffic flow forecasting with existing incomplete data [8]. In 2008, a technique was proposed by Yang Zhang, which named least squares support vector machines (LS-SVMs) and it's a non-parametric method, Zhang held that the method had the ability of great generalization, which was better than others, and the result about the traffic flow forecasting was effective in reducing errors, and it was proved that the method was a hopeful technique in traffic forecasting [9]. Senyan Yang used a method called Gradient boosting decision trees (GBDT) to forecast the traffic flow, which is based on the data of traffic flow gathered by loop detectors. It could improve prediction performance by imputing the collected traffic flow data and then processing it, and this method got good performance and model interpretability for traffic flow forecasting [10]. E.Y.Kim proposed a model based on the 3D Markov random field (MRF) for traffic flow forecasting, and he made use of the clique method to represent the relationship among roads at a given location and determined its structure, which improved the performance compared with the state-of-the-art technique[11].

Although these methods all achieved satisfactory performance on their datasets; however, the performance cannot be generalized to different scenes. Besides, some traditional machine learning methods cannot capture the correlation between high-dimensional features, thus causing the performance to decrease. Therefore, it is necessary to develop a deep learning-based method to further extract spatial information and improve the prediction performance.

III. DATASET

In this study, we utilize a publicly available dataset for evaluation. The dataset is downloaded from https://www.kaggle.com/jboysen/us-traffic-2015, which includes 2 files, and before the final model construction, we conduct some data cleaning steps and data preprocessing. First, we delete the missing values in the dataset, and then list the feature values we need, filter the data to what we need, mark the location of the station on the map, and check the data type of the station and the digital features of the dataset. Next, we throw away unnecessary data columns and ensure that we have evenly distributed data throughout the year. After deleting invalid data, we import the traffic flow dataset and observe the histogram of each variable.

Besides, we read the holiday data of the United States from https://gist.github.com/shivaas/4758439 and import the holiday data into the traffic flow dataset, and then clear up the traffic data affected by the holidays, discard the outliers, and view the linear correlation and correlation matrix and fit the data to get the median value of each column. Taking August as an example, first, we find the average weather data for August, and fill in the missing data to reshape the one-dimensional matrix into a two-dimensional matrix, introduce the 'precip' variable, replace the missing value with the median of each column, calibrate all weather-related features, connect the median of each row, and get the final predicted value. After these data processing and feature engineer steps, the processed features and corresponding descriptions can be seen in Table 1.

Feature id	Feature description
date	date of that day
day_of_data	The order of that day in that month
day_of_week	The order of that day in that week
direction_of_travel	The direction of travel for traffic flow of that day
month_of_data	The month the day is in
conds	The weather condition of that day
rain	Whether it was raining that day
snow	Whether it was snowing that day
Tempi	The temperature of that day
visi	Visibility of that day
holiday_flag	whether that day a bank holiday

Table 1. Feature description for traffic flow prediction

IV. METHOD

In this study, we select to use CNN for our regression task. CNN is good at image processing and has two major characteristics: 1) It can effectively reduce the dimensionality of a large amount of data into a small amount of data and will not have a major impact on the result; 2) It can effectively retain the image features. The problems that these two major features solve respectively are: 1) The amount of data that the image needs to be processed is too large, resulting in high costs and low efficiency; 2) It is difficult for the image to retain the original features during the digitization process, resulting in low accuracy rate in image processing. CNN uses a visual-like method to retain image features, no matter how the image changes position, CNN can accurately identify similar images. The principle similar to human vision used in CNN is probably that the original signal is ingested, the signal is initially processed, and then abstracted and further processed [12].

Basically, CNN consists of three parts: convolutional layer; pooling layer; and fully connected layer. The main function of the convolutional layer is to retain the characteristics of the picture. The process of extracting features by the convolutional layer can be understood as using a filter to filter small areas of the image to obtain the feature values of these small areas. In practical applications, there are often multiple convolution kernels, and different convolution kernels represent different image modes. If a certain image block and the convolution kernel convolve a large value, it is considered that the image block may have important features [13].

Convolution is one-dimensional convolution and multi-dimensional convolution. This article takes two-dimensional convolution as an example to illustrate the difference between one-dimensional convolution and two-dimensional convolution. The first is the difference in dimension. One-dimensional convolution only needs the input and output in a three-dimensional format (batch, channel, inputDim), two-dimensional convolution requires four-dimensional input and output (batch, channel, inputH, inputW)[14]. Another difference is followed by the calculation, one-dimensional convolution only needs to be calculated in the dimension of inputDim, while two-dimensional convolution needs to be calculated in both dimensions of input H and input W at the same time. There are also differences in the meaning of one-dimensional and two-dimensional convolution. A picture of one-dimensional convolution is a vector, and two-dimensional convolution has width and height. The object of the operation is the dimension, so the input of the two-dimensional convolution is the real picture [15]. The formula of one-dimensional convolution can be defined as Equation (1).

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) g(x-\tau) d\tau \tag{1}$$

where *f(x)*g(x)* represents the convolution of *f(x)* and *g(x)*, the independent variable is *x*. The formula of two-dimensional convolution can be defined as Equation (2).

$$f(x,y) * g(x,y) = \int_{\tau_1=-\infty}^{\infty} \int_{\tau_2=-\infty}^{\infty} f(\tau_1,\tau_2) \cdot g(x-\tau_1, y-\tau_2) d\tau_1 \tau_2 \tag{2}$$

However, most of the discrete forms of two-dimensional convolution which can be defined as Equation (3) are used in image processing:

$$f[x,y]*g[x,y]=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} f[n_1,n_2]\cdot g[x-n_1,y-n_2] \quad (3)$$

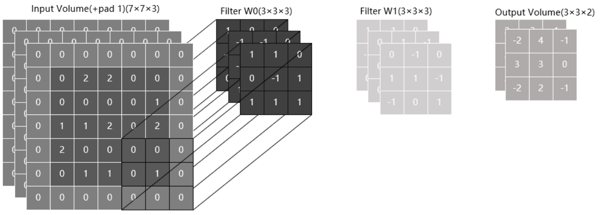

Figure 1. Convolution calculation process

Figure 1 is an example of the convolution calculation process. The input image size is *H=5, W=5, D=3*, which is a three-channel color image with a size of 5*5. The figure contains two sets of convolution kernels, namely Filter W0 and W1. In the convolution calculation, different convolution kernels are used for different input channels. In this example, the sliding step length of the convolution kernel in the horizontal and vertical directions of the image is 2, and zero is filled around the input image. After convolution, the output is a 3*3*2 feature map, that is, a 2-channel feature map with a size of 3*3. And each pixel in the output feature map is the summation of the inner product of each set of filters and each feature map of the input image, and the offset b0 is added. For different output feature maps, the offset is the same. The last feature value in the output feature map o [:, :, 0] can be defined as Equation (4).

$$Value\ o[2,2,0]=\sum x[:,:,0]\times w[:,:,0] + \sum x[:,:,1]\times w[:,:,1]+\sum x[:,:,2]\times w[:,:,2]+b_0 = -2 \quad (4)$$

The main function of the pooling layer is to reduce the dimensionality of the data and effectively avoid overfitting. Since the convolution kernel is small, the image after the convolution is still large, and the function of the pooling layer is to perform down-sampling to reduce the dimensionality of the data. Compared with the convolutional layer, the pooling layer can more effectively reduce the data dimension, not only can reduce the amount of calculation, but also effectively prevent overfitting [16].

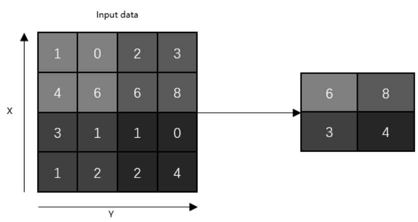

Figure 2. The calculation process of the pooling layer

In the pooling layer, if the output matrix size is x, the input matrix size is n, the convolution kernel size is f, and the step size is s, the calculation process of the pooling layer can be seen in Figure 2, then the output matrix size of the pooling layer can be defined as Equation (5).

$$x=\frac{n-f}{s}+1 \quad (5)$$

If the obtained value of x is a decimal, then the value is taken down. The function of the fully connected layer is to output the results we want according to different tasks. Input the reduced data to the fully connected layer to obtain the final desired result. A typical CNN is not only a three-layer structure. For example, the structure of LeNet-5 is a six-layer structure of the convolutional layer-pooling layer-convolutional layer-pooling layer-convolutional layer-fully connected layer [17].

Taking LeNet-5 as an example, we first normalize the input image to 32*32, performs the first convolution operation on the input image in the C1 convolution layer, and obtain 6 28*28 C1 feature maps, and then in the S2 pool The transformation layer performs pooling operation, using 2*2 cores for pooling, to get 6 14*14 feature maps, and then convolution in the C3 convolution layer to get 16 10*10 feature maps, the convolution kernel used The size of is 5*5, and then in the S4 pooling layer, the 16 10*10 of the C3 layer are pooled in 2*2 units to obtain 16 5*5 feature maps, and the convolution kernel The size is the same, so the size of the image formed after convolution in the C5 convolution layer is 1*1, and there are 120 convolution results. In the fully connected layer, 84 nodes are obtained, corresponding to a 7*12 bitmap, and finally, 10 nodes are obtained in the OUTPUT layer, representing 0 to 9, respectively. If the value of node i is 0, the identified result is a number. Using the RBF network connection method, assuming that x is the input of the upper layer and y is the output of RBF, the RBF output calculation formula can be defined as Equation (6).

$$y_i=\sum_j\left(x_j-w_{ij}\right)^2 \quad (6)$$

Here we use a one-dimensional CNN for our task and it can be seen in Figure 3. Firstly, we reshape the image, and then the first convolution operation is carried out on the reshaped image. The next step is to perform Avg-pool on the image, and then we convolve the image three times, after that, performing Max-pool on the image. Finally, reshaping the image and getting the result.

V. METRIC

Since in our study the prediction of traffic flow is a regression task, therefore, we use some common metrics in regression, which are *RMSE, MAE*, and R^2. The *RMSE* is the arithmetic

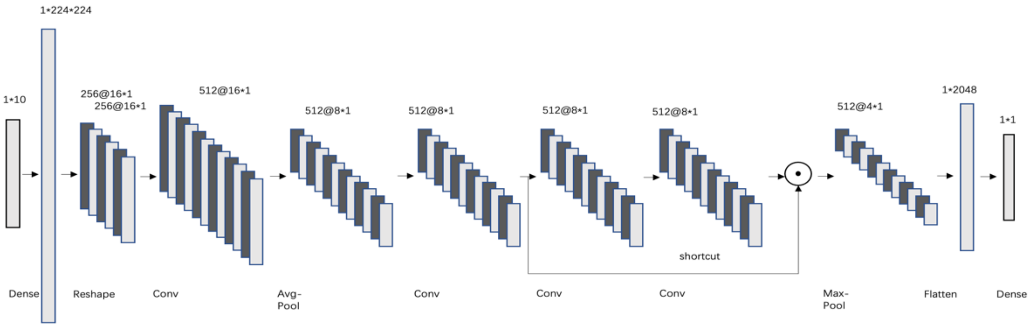

Figure 3. The process of a one-dimensional CNN

square root of the mean square error, which is used to measure the deviation between the observed value and the true value and can well reflect the precision of measurement.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f_i - y_i)^2} \tag{7}$$

The variables of *RMSE* are m, f_i, and y_i, and the significance of *RMSE* is that when you take the square root, the error is on the same level as the data, so you can better describe the data [18].

The variables of *MAE* are m, f_i, and y_i. The *MAE* is the average value of absolute error, which can reflect the actual situation of predicted value error. The calculation formula of *MAE* can be defined as Equation (8).

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|f_i - y_i| \tag{8}$$

The calculation formula of *R-squared* can be defined as Equation (9).

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(f_i - y_i)^2}{\sum_{i=1}^{m}(\bar{y}_i - y_i)^2} \tag{9}$$

The variables of R^2 are m, f_i, and y_i. The numerator is the sum of the error predicted by the model trained by us, and the denominator is the sum of the error estimated by us (usually taking the average value of predictions). The value of R^2 is between 0 and 1, and the closer it is to 1, the better the regression fitting effect is. It is generally believed that the model over 0.8 has higher goodness of fit [19]. Herein, m is the number of instances in the dataset that you use to measure; f_i is the value we observe; y_i is the real value.

Besides, we run the experiments based on 10-fold cross-validation. Firstly, dividing the data into ten parts, then taking turns using 9 of them as training data and 1 of them as test data. Several tests will get several correct rates or error rates, and the average of the correct rate or error rate of 10 times is used as the estimate of the accuracy of the algorithm, and we use multiple 10-fold cross-validations to find the average values as the algorithm accuracy of estimates [20].

VI. RESULT

Metric / Method	RMSE (STD)	MAE (STD)	R-squared (STD)
Logistic Regression	0.2777 (0.0220)	0.2432 （0.0171）	0.0427 (0.0892)
Support Vector Machine	0.3059 （0.0360）	0.2266 （0.0310）	-0.1636 (0.1909)
Bayes Classifier	0.2769 (0.0183)	0.2480 (0.0148)	0.0487 (0.0598)
Decision Tree	0.2084 (0.0409)	0.1234 (0.0339)	0.4457 (0.2059)
Neural Network	0.2105 (0.0334)	0.1593 (0.0204)	0.4390 (0.1806)
1D-CNN (Ours)	0.2001 (0.0374	0.1494 (0.0285)	0.4919 (0.1841)

Table 2. Results of several methods and metrics

The calculation formula of *RMSE* can be defined as Equation (7).

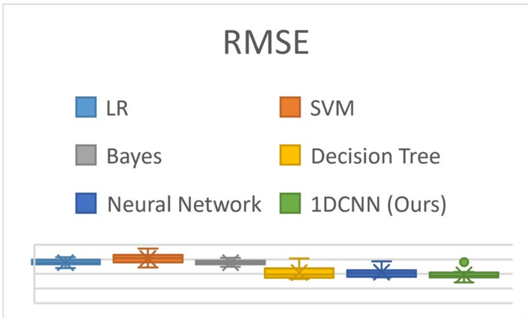

Figure 4. Result of RMSE

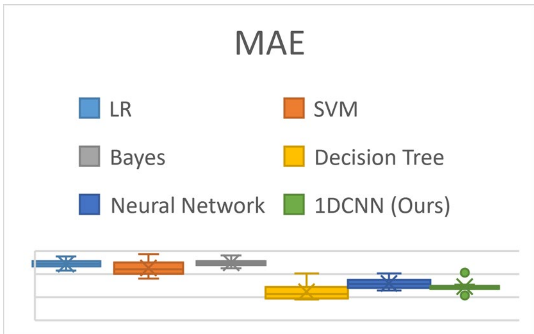

Figure 5. Result of MAE

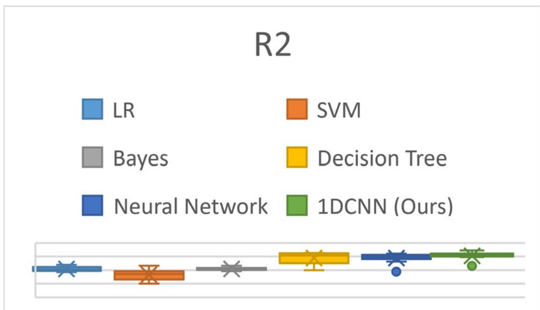

Figure 6. Result of R^2

The experiment results can be seen in Figures 4, 5, 6, and Table 2. As shown in Table 2, the average value of *RMSE* of the 1DCNN method selected by the author is the lowest compared with other methods, that is, the 1DCNN method has the smallest error. Bayes method has the lowest variance. As shown in the boxplot, the graph of 1DCNN is in a lower position on the Y-axis as a whole, and the lower the *RMSE* value, the better this method is. For *MAE*, the Decision Tree has the lowest value, that is, the method error of the Decision Tree is the smallest. The 1DCNN method used in this paper has the second smallest error, and its variance is lower than that of the Decision Tree, that is, the 1DCNN method is more stable than the Decision Tree. In addition, the *MAE* value obtained by the 1DCNN method is relatively low, which proves that the 1DCNN method has a good effect. For *R2*, the larger the value is, the smaller the error of this method will be. The graph of 1DCNN in the figure is above the Y-axis as a whole.

VII. CONCLUSION

The versatility of deep learning enables it to be used in many fields, such as computer vision, automatic driving, and other fields and has achieved success [21]. Moreover, deep learning has a very advanced performance. The larger the number of samples, the higher the accuracy of deep learning and probably better than the traditional machine learning algorithm, so deep learning has a very important significance in the field of traffic flow prediction. However, deep learning also has its shortcomings [22]. For example, deep learning has high requirements on data, and most mainstream deep learning frameworks are expensive. Parameter tuning in deep learning is essential, but the process is too slow. In the future, the accuracy can be improved by combining CNN's ability to process image data with LSTM's ability to extract features, and deep learning can be applied to ITS. Meanwhile, in order to solve the overfitting phenomenon, the problem can be solved by applying dropout to ignore some parameter updates or using regularization of L1 or L2 norm.

REFERENCES

[1] Jeong Y S, Byon Y J, Castro-Neto M M, et al. Supervised weighting-online learning algorithm for short-term traffic flow prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(4): 1700-1707.

[2] Luo C, Huang C, Cao J, et al. Short-term traffic flow prediction based on least square support vector machine with hybrid optimization algorithm[J]. Neural processing letters, 2019, 50(3): 2305-2322.

[3] Williams B M, Hoel L A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results[J]. Journal of transportation engineering, 2003, 129(6): 664-672.

[4] Ma X, Dai Z, He Z, et al. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction[J]. Sensors, 2017, 17(4): 818.

[5] Moses A, Parvathi R. Vehicular traffic analysis and prediction using machine learning algorithms[C]//2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE, 2020: 1-4.

[6] Sun Y, Leng B, Guan W. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system[J]. Neurocomputing, 2015, 166: 109-121.

[7] Luo X, Li D, Yang Y, et al. Spatiotemporal traffic flow prediction with KNN and LSTM[J]. Journal of Advanced Transportation, 2019, 2019.

[8] Sun S, Zhang C, Yu G. A Bayesian network approach to traffic flow forecasting[J]. IEEE Transactions on intelligent transportation systems, 2006, 7(1): 124-132.

[9] Zhang Y, Liu Y. Traffic forecasting using least squares support vector machines[J]. Transportmetrica, 2009, 5(3): 193-213.

[10] Yang S, Wu J, Du Y, et al. Ensemble learning for short-term traffic prediction based on gradient boosting machine[J]. Journal of Sensors, 2017, 2017.

[11] Kim E Y. MRF model based real-time traffic flow prediction with support vector regression[J]. Electronics Letters, 2017, 53(4): 243-245.

[12] Yu D, Yang L, Yu X. A Data Grouping CNN Algorithm for Short-Term Traffic Flow Forecasting[C]// Asia-Pacific Web Conference. Springer International Publishing, 2016.

[13] Sun, Carlos, Edara, et al. Traffic Flow Forecasting for Urban Work Zones[J]. IEEE transactions on intelligent transportation systems, 2015, 16(4):1761-1770.

[14] Qiao Y, Wang Y, Ma C, et al. Short-term traffic flow prediction based on 1DCNN-LSTM neural network structure[J]. Modern Physics Letters B, 2020(3):2150042.

[15] Castillo E, JM Menéndez, S Sánchez-Cambronero. Predicting traffic flow using Bayesian networks[J]. Transportation Research Part B Methodological, 2008, 42(5):482-509.

[16] Ding A L, Zhao X M, Jiao L C. Traffic flow time series prediction based on statistics learning theory[C]// Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on. IEEE, 2002.

[17] Messai N, Thomas P, Lefebvre D, et al. A neural network approach for freeway traffic flow prediction[C]// Control Applications, 2002. Proceedings of the 2002 International Conference on. IEEE, 2002.

[18] Yu Y J, Cho M G. A Short-Term Prediction Model for Forecasting Traffic Information Using Bayesian Network. IEEE Computer Society, 2008.

[19] Rousson V, Gosoniu N F. An R -square coefficient based on final prediction error[J]. Statistical Methodology, 2007, 4(3):331-340.

[20] Hamed M G, Gianazza D, Serrurier M, et al.: United States (2013)" Statistical prediction of aircraft trajectory: regression methods vs point-mass model. 2013.

[21] Polson N G, Sokolov V O. Deep learning for short-term traffic flow prediction[J]. Transportation Research Part C Emerging Technologies, 2017, 79(JUN.):1-17.

[22] Wang Y, Zhang D, Liu Y, et al. Enhancing transportation systems via deep learning: A survey[J]. Transportation Research, 2019, 99(FEB.):144-163.