ISBN 978-981-14-4787-7
Proceedings of 2020 the 10th International Workshop on Computer Science and Engineering (WCSE 2020)
Shanghai, 19-21 June, 2020, pp. 52-56
doi: 10.18178/wcse.2020.06.009

Traffic Congestion Analysis Based on Deep Neural Networks

Di Zang [+], Xiaoke Qu and Yang Fang

Department of Computer Science and Technology, Tongji University, Shanghai, China

Abstract. Traffic congestion analysis is a research hotspot in intelligent transportation system (ITS). However, existing research methods focus on real-time or short-term traffic congestion determination and prediction. As the time interval increases, noise and sudden abnormalities contained in traffic data increase. How to predict congestion state throughout the next day is still a challenging task that has not been well solved. In this paper, we propose a multi-branch congestion model (MC) based on the multi-branch speed prediction model (MSP) to realize long-term traffic congestion prediction. Experiments based on Shanghai elevated highways show that our method is robust and has better performance.

Keywords: traffic congestion analysis, speed/congestion prediction, deep neural networks

1. Introduction

To improve traffic conditions, predicting future traffic congestion state is an important subject in the field of intelligent transportation system research. The purpose of traffic congestion analysis is to dig out the potential laws of traffic congestion changes in complex historical traffic data, such as vehicle speed, and to achieve prediction and determination of the future congestion state. Now, many methods have been proposed to determine traffic congestion state. Based on numerical statistical analysis of different traffic parameters, Jiang et al. [1] and Sobral et al. [2] proposed two methods of traffic congestion classification. Traditional machine learning algorithms are also used to classify and predict traffic congestion, such as k-means clustering algorithm [3], support vector machine (SVM) [4], random forest [5], etc. At present, researchers tend to use deep learning methods to achieve traffic congestion prediction [6] [7].

However, some of the above methods lack analysis of massive traffic data, and some only achieve real-time or short-term prediction. Based on historical speed data, we propose the MSP/MC model to predict traffic congestion state throughout the next day.

2. Proposed Methodology

2.1. Overview

Our system architecture is shown in Figure 1. Firstly, raw data is pre-processed and cleaned to remove abnormal data, and then transformed into spatial-temporal matrices. Next, we randomly divide the samples to train and predict. Then, we first propose the MSP model to achieve long-term traffic speed prediction. Based on the MSP model with excellent performance, the MC model for long-term congestion prediction is then proposed, which will also have good performance.

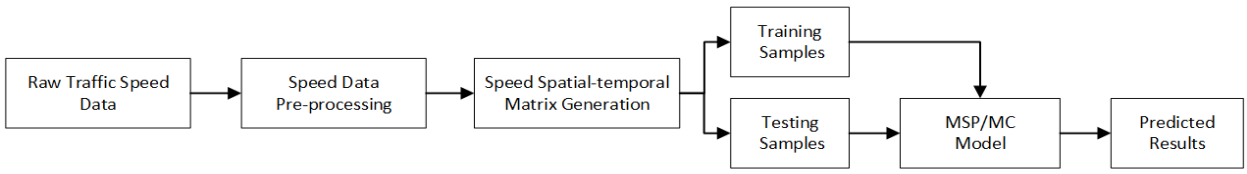

Fig. 1: The system architecture of traffic speed and congestion prediction using the MSP/MC model.

[+] Corresponding author.
E-mail address: zangdi@tongji.edu.cn.

2.2. Traffic Speed Spatial-temporal Matrix Generation

Traffic speed data is collected by detectors deployed on the road. The detectors are spaced at equal distances and collect traffic data at regular intervals. In order to make full use of the correlation between time and position, we construct the traffic speed spatial-temporal matrix which can be expressed as Eq. (1):

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

The matrix X represents the traffic speed information for a day. Where m and n are the number of loop detectors and the number of time intervals respectively, x_{ij} is the average speed of the ith loop detector at the jth time period.

2.3. The MSP Model for Traffic Speed Prediction

To learn more robust features, we propose a personalized design model called MSP as demonstrated in figure 2. Taking into account the time correlation among days, our model uses a three-branch structure, taking three historical speed matrices as input, to predict the traffic speed on the Tth day. Inputs include the speed matrix of the day before predicted day and the speed matrices of the same weekdays as predicted day in the previous two weeks. Each branch uses a CRDNN layer to extract the spatial and temporal features of a historical day. Then integrate the characteristics of the three historical days. Finally, a fully connected layer is used, which decodes the extracted abstract features and outputs the final speed prediction result.

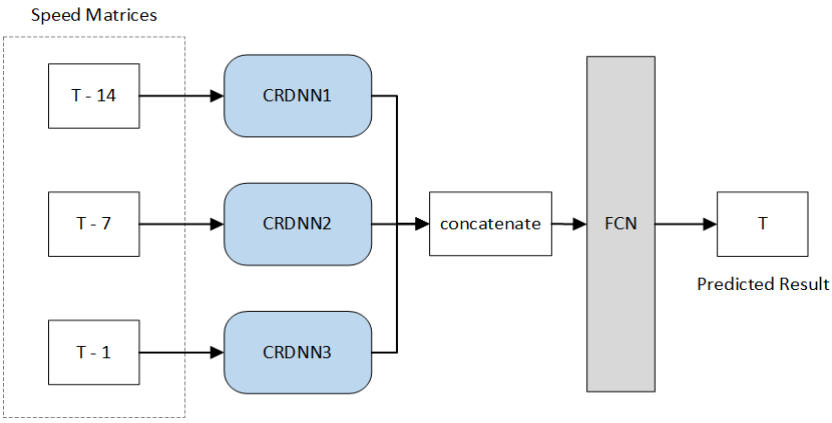

Fig. 2: The structure of the MSP model.

The CRDNN Layer. Figure 3 shows the structure of a CRDNN layer. The CRDNN layer uses convolution layers to extract abstract features contained in the historical speed spatial-temporal matrices, and uses deconvolution layers to re-decode the learned abstract features into predicted speed matrix. Then, a residual structure is introduced in the entire network, which can directly jump information of the previous layer to the input of the later layer. The residual structure not only solves the problem of difficult training of multilayer neural network, but also improves the integrity of the original speed information.

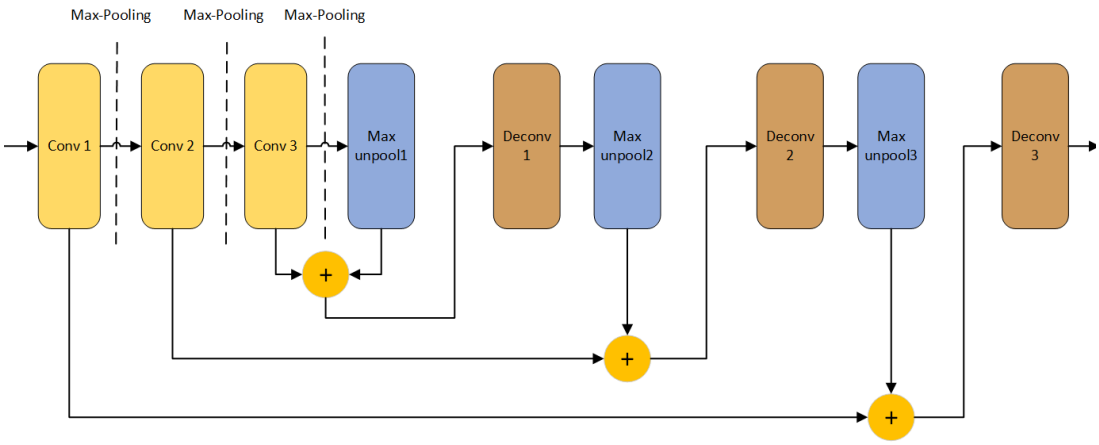

Fig. 3: The structure of the CRDNN layer.

Training Optimization for Regression Tasks. Traffic speed prediction is essentially a regression problem. Therefore, here we use the mean-square error as the loss function of the MSP model, which is optimized by Adam optimizer. Our loss function is calculated as Eq. (2):

$$Loss_{MSP} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}^{(i)} - y^{(i)})^2 \tag{2}$$

where N represents the number of data, $\hat{y}$ represents the prediction by our MSP model, and y represents the ground truth.

2.4. The MC Model for Traffic Congestion Prediction

In the field of transportation, an important factor for determining congestion level is speed. Based on the speed prediction model (MSP) with excellent performance, the proposed MC model will also have excellent performance in congestion state prediction. It is worth noting that the MSP model is used for traffic speed data prediction. Traffic speed data prediction is a regression task, and the output data is continuous. However, Traffic congestion is usually divided into three levels which are unblocked, lightly congested, and severely congested. Traffic congestion prediction is a classification task that requires discrete labels, and its output is discrete.

Discrete Labels Generation. At present, there is no unified standard for the definition of traffic congestion level in various countries. Thus, we use the unsupervised clustering algorithm K-means to automatically classify traffic congestion levels based on the numerical characteristics of the dataset itself. First, cluster the speed values in the speed spatial-temporal matrix. Then determine the number of clustering centers according to the number of congestion levels you want to divide, and classify each point into the class of the clustering center closest to it. Next calculate the proportion of points in each category to the whole. Finally, the category with the largest proportion is selected as a full-day traffic congestion level label.

The MC Model. Different from the MSP model (Fig. 2), the input labels of the MC model are the numbers 0, 1, and 2, respectively, corresponding to three types of traffic congestion states which are unblocked, lightly congested, and severely congested. In addition, because congestion prediction is a classification task, a softmax layer is added behind FCN layer at the end. The number of neurons is 3, and the probability that the congestion status on the prediction day belongs to three categories is output.

Training Optimization for Classification Tasks. We choose cross entropy as the loss function, which is commonly used in classification problems. Our loss function is calculated as Eq. (3):

$$Loss_{MC} = -\frac{1}{N}\sum_{x}(y ln\hat{y} + (1-y)ln(1-\hat{y})) \tag{3}$$

where N represents the number of data, $\hat{y}$ represents the prediction by our MC model, and y represents the real label.

3. Experiment and Results

3.1. Datasets and Settings

The real speed data is collected from detectors deployed on Yan'an and Neihuan elevated highways in Shanghai, China throughout 2011. The two elevated highways are important parts of Shanghai transportation network. On the elevated highway, there exists a loop detector every 400 meters that records speed values every five minutes. The number of detectors deployed in Yan'an and Neihuan elevated highways are 35 and 72, respectively. On Yan'an elevated highway, because traffic system upgrades or detector failures, there is four-day data missing from March 20 to March 23, and the data available for research is 361 days. After repairing the abnormal data, we find that the night traffic mode is single and generally unobstructed, so we choose the data collected from 7 am to 22 pm for analysis. For each elevated highway, 30 samples are randomly selected as the test set, and the rest are used as the training set. Data augmentation is used.

The experiments are conducted on a server with i7-5820K CPU, 48GB memory and NVIDIA GeForce GTX1080 GPU. The proposed models are implemented on the TensorFlow framework of deep learning. The parameter configuration of our proposed models for 2 elevated highways are shown in table 1. To verify the robustness of our models, predictions on two elevated highways use models with the same parameter configuration. In addition, the parameter configurations of three CRDNN branches are also the same.

Table 1: Parameter configuration of the MSP/MC model.

Layers	Name	Description
CRDNN	Conv1	Kernel number = 64; Kernel size = 5*5*1
CRDNN	Max-Pooling1	Kernel size = 2*2; Stride = 2
CRDNN	Conv2	Kernel number = 64; Kernel size = 3*3*64
CRDNN	Max-Pooling2	Kernel size = 2*2; Stride = 2
CRDNN	Conv3	Kernel number = 128; Kernel size = 3*3*64
CRDNN	Max-Pooling3	Kernel size = 2*2; Stride = 2
CRDNN	Max-Unpooling1	Kernel size = 2*2; Stride = 2
CRDNN	Deconv1	Kernel size = 3*3; output shape = output shape of Max-Pooling2
CRDNN	Max-Unpooling2	Kernel size = 2*2; Stride = 2
CRDNN	Deconv2	Kernel size = 3*3; output shape = output shape of Max-Pooling1
CRDNN	Max-Unpooling3	Kernel size = 2*2; Stride = 2
CRDNN	Deconv3	Kernel size = 5*5; output shape = shape of input
FCN	FCN	Neuron nodes number = Flatten input

3.2. Traffic Speed Prediction Results

In experiments, we compared the proposed model (MSP) with traditional convolutional neural networks. The single-branch CNN only uses the speed data of the previous day as input. The three-branch CNN has the same inputs as the model we proposed, using historical three-day data. To evaluate the performance of models, we use 3 criteria which are mean relative error (MRE), mean absolute error (MAE), and root mean square error (RMSE). The calculation methods of MRE, MAE and RMSE are given by Eq. (4)-(6):

$$\mathrm{MRE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}^{(i)}-y^{(i)}|}{y^{(i)}} \tag{4}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}^{(i)}-y^{(i)}| \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}^{(i)}-y^{(i)})^2} \tag{6}$$

The comparative experimental results of Yan'an and Neihuan elevated highways are shown in table 2. From the table, we can see that our model can obtain better prediction results on both elevated highways.

Table 2: Comparative experimental results of two elevated highways.

Models	Yan'an elevated highway			Neihuan elevated highway		
	MRE	MAE	RMSE	MRE	MAE	RMSE
Our model	0. 17523	6. 98647	10. 45069	0. 15960	6. 01115	9. 63461
Three-branch CNN	0. 19944	7. 77331	11. 67639	0. 17935	7. 49362	11. 41411
Single-branch CNN	0. 20844	8. 15142	11. 92533	0. 18997	7. 67832	11. 73552

In order to reflect the fitting degree between real values and predicted values, we also randomly select two detectors from the test samples of the two elevated highways, and make the change curves of their predicted and real values. The results are shown in figure 4. The x-axis denotes the time series of a day, and the y-axis indicates the values of speed. The curves show that at a certain location, the predicted value of speed for a whole day is basically the same as the true value.

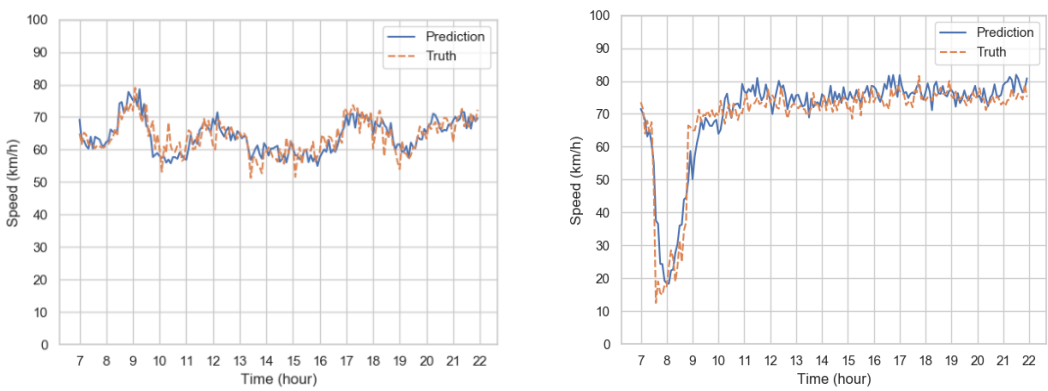

Fig. 4: Speed change curves of Yan'an (left) and Neihuan (right) elevated highways.

3.3. Traffic Congestion Prediction Results

As mentioned above, first we determine the thresholds for dividing different congestion levels. The speed values of three cluster centers on Yan'an elevated highway are 80.116325 (km/h, unblocked), 61.235867 (km/h, lightly congested), and 30.388401 (km/h, severely congested), while the speed values on Neihuan elevated highway are 74.93975, 59.669064, and 18.136627. Different roads have different thresholds for different congestion levels. It's also confirmed that a fixed speed value cannot be used to classify the congestion status of different roads. Then, we can easily label the traffic congestion of a day. After labelling data, use the MC model to learn. Single-branch CNN and three-branch CNN are also used as comparison models. The comparative experimental results of Yan'an and Neihuan elevated highways are shown in table 3. From the table, it can be seen that our model is more accurate in predicting traffic congestion state.

Table 3: Classification results of traffic congestion level on two elevated highways.

Models	Accuracy	
	Yan'an	Neihuan
Our model	93. 3%	93. 3%
Three-branch CNN	83. 3%	90. 0%
Single-branch CNN	80. 0%	86. 7%

4. Conclusions

In this paper, we propose a multi-branch congestion model (MC) based on a multi-branch speed prediction model (MSP) to handle the challenging task of long-term traffic congestion situation prediction. The MSP model takes advantages of residual and deconvolution theories while simultaneously learns multiple inputs in both the spatial and temporal domains. It successfully digs out potential non-linear traffic speed features and achieves long-term traffic speed prediction. Based on the effective MSP model, the MC model is further proposed to predict the long-term congestion state. Its input is labeled by an efficient adaptive method. Tested on real data, experimental results illustrate that our models are robust and can gain better prediction results.

5. Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 61876218).

6. References

[1] Jiang G, Jiang G, NiuS, et al. Automatic Traffic Congestion Identification Method of Expressway based on Gain Amplifier Theory[C]. International Conference on Advanced Computer Control (ICACC 2010), 2010: 648-651.

[2] Sobral A, Oliveira L, Schnitman L, et al. Highway Traffic Congestion Classification using Holistic Properties[C]. 10th IASTED International Conference on Signal Processing, Pattern Recognition and Applications, 2013: 458-465.

[3] Wei L, Li M. A fuzzy fusion algorithm to identify the traffic states by the floating car data. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012:5-9.

[4] Hong Nam Nguyen, Krishnakumari P, Vu H. L, et al. Traffic Congestion pattern classification using multi-class SVM. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016:1059-1064.

[5] Liu Y, Wu H. Prediction of Road Traffic Congestion Based on Random Forest[C]. 2017 10th International Symposium on Computational Intelligence and Design (ISCID), 2017: 361-364.

[6] Chen Y Y, Lv Y, Li Z, et al. Long short-term memory model for traffic congestion prediction with online open data[C]. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016: 132-7.

[7] Wang P, Li L, Jin Y, et al. Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a CNN-architecture trafficnet. 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2018:1134-1139.