

## Research Article

# A Deep Neural Network Based on Classification of Traffic Volume for Short-Term Forecasting

Jing Bai  and Yehua Chen

*Yanshan University School of Economics and Management, 066004 Qinhuangdao, China*

Correspondence should be addressed to Jing Bai; 13383351868@163.com

Received 9 October 2018; Revised 8 January 2019; Accepted 5 February 2019; Published 20 February 2019

Academic Editor: Alexander Paz

Copyright © 2019 Jing Bai and Yehua Chen. 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.

This paper developed a deep architecture to predict the short-term traffic flow in an urban traffic network. The architecture consists of three main modules: a pretraining module, which generates initialized weights and provides a rough learning of the features firstly with the training set in an unsupervised manner; a classification module, which performs the data classification operation through adding the logistic regression on top of the pretrained architecture to distinguish the traffic state; and a fine-tuning module, which predicts the traffic flow with supervised training based on the initialized weights in the first module. The classification module provides the fine-tuning modules with two classified datasets for more accurate forecasting. Furthermore, both upstream and downstream data are utilized to improve the prediction performance. The effectiveness of the proposed model was verified by the traffic prediction of the road segments of Nanming District of Guiyang. And with the comparison analysis over the existing approaches, the proposed model shows superiority in short-term traffic prediction, especially under incident conditions.

## 1. Introduction

Short-term traffic flow forecasting is important for the study of intelligent transportation systems. The accurate and timely traffic flow forecasting can effectively relieve the traffic congestion, reduce the incidence of accidents, and provide us comfortable traffic environment. The short-term traffic flow prediction, whose time span is generally not more than 15 minutes, makes the influence of random factors and uncertainty more significant. So an adaptive prediction model with the ability of dealing with time series data and fast computing ability is highly overarching.

Traditional traffic flow prediction approaches can be divided into two categories: prediction method which is based on traditional mathematics and physics, such as ARIMA model [1], time series model [2], Kalman filtering model, and exponential smoothing model. But it is hard to actually apply these prediction methods to make timely and accurate prediction, due to the difficulty in constructing and solving the mathematical model utilized in these methods. The other kind is the approaches without mathematical models, including neural networks [3], nonparametric regression

[4], and support vector machines (SVM) [5, 6] which do not need to build a complex model and only utilizes real-world dataset to make predictions. In recent years, the breakthrough in deep learning algorithms has led to its application in transportation systems which have large amounts of data with large dimensions. In [7], a deep Restricted Boltzmann Machine and Recurrent Neural Network architecture was utilized in the model, and the traffic congestion was predicted based on Global Positioning System (GPS) data from taxi. In order to obtain better prediction accuracy for intelligent transportation systems, the preprocess on the raw data in advance is proposed. In [8], a hybrid model for traffic flow prediction was developed, the FIFO-filter classified data into clusters and made a rough prediction, and the multilayer feed-forward neural network architecture was optimized using evolutionary strategies to provide accurate prediction. In [9] a neurofuzzy model was employed for traffic prediction. It used a gate network to categorize the input data based on a fuzzy approach and an expert neural network to specify the input-output relationship.

Traffic flow datasets have significant characteristics; for example, it has both time dimension and space dimension

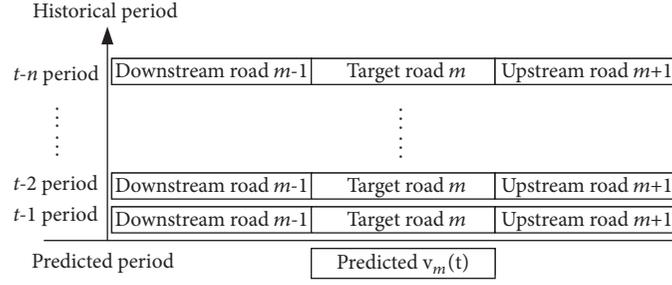


FIGURE 1: Forecasting framework.

and it could be influenced by traffic accidents, weather, festival seasons, etc. In [10] the spatiotemporal characteristics of traffic flow were considered, and the spatial and temporal correlation was introduced into the modeling phase to improve prediction accuracy. Reference [11] forecasted normal traffic flow in traditional way, while at the edges of the rush hours, the autoregression algorithm combined with historical data was utilized.

Based on the preliminary investigation, a certain characteristic of traffic flow is identified such that there would be a big change in the relationship between upstream traffic parameters and downstream traffic parameters if the traffic accidents occurred. Incorporated with this characteristic and making full use of huge real-world data, this paper proposed a deep architecture consisting of three modules for traffic flow prediction: in the first module, we attempt to train a deep belief network with stacked Restricted Boltzmann Machines. The trained weights of DBN have two roles; one is as features extraction of traffic state classification. Another is used for unsupervised features learning in traffic flow prediction. In the second module, the training data were divided into two datasets (the normal and the accidents). Then the datasets were used to train the classifier. The division could contribute to proper training and reduce the high error caused by spatial difference between different traffic conditions. In the third module, supervised training based on the initialized weights obtained from the first module was developed combined with each classified dataset to predict the traffic flow respectively. The respective fine-tuning could capture the unique features of two conditions and help to have a faster convergence. The performance of the proposed model was compared with conventional neural networks. The rest of the paper is structured into three sections. In Section 2, a brief introduction on the traffic prediction problem and the proposed model architecture is presented. Section 3 provides detail explanation of the test data used, the selection of various parameters, and benchmark models used and details the simulation results. Furthermore, the comparison and analysis of the performance of proposed prediction model against benchmark models are provided. Section 4 provides a brief conclusion.

## 2. Traffic Flow Prediction

**2.1. Basic Description.** Traffic flow can be affected by previous traffic conditions in the upstream and downstream direction.

Assume that a target road section is denoted as  $m$  and the forecasted period as  $t$ . In order to predict the average speed of vehicles on road  $m$  at time  $t$  which is denoted as  $v_m(t)$ , the forecasting framework is built as shown in Figure 1.

There is a certain relationship between  $v_m(t)$  and the average speed of the vehicles during the previous period from the temporal dimension. And  $v_m(t)$  is affected by the traffic flow of the upstream and downstream sections simultaneously. As a result,  $v_m(t)$  can be predicted by  $n$  previous period  $v_m(t-1), v_m(t-2), \dots, v_m(t-n)$  and traffic state of upstream and downstream sections during the same period  $v_{m-1}(t), v_{m+1}(t)$ . Similarly, traffic flow of other segments can also use this framework to predict. Let the input matrix of road network state information be  $X$ ; the rows of input matrix consist of traffic flow data which is acquired orderly according to the time. Each column is the traffic data of road  $m-1, m, m+1$ . Each row is a high-dimensional sample with a combination of column information. The input matrix is expressed as

$$X = \begin{bmatrix} V_{m-1}(t-1) & V_m(t-1) & V_{m+1}(t-1) \\ V_{m-1}(t-2) & V_m(t-2) & V_{m+1}(t-2) \\ \vdots & \vdots & \vdots \\ V_{m-1}(t-N) & V_m(t-N) & V_{m+1}(t-N) \end{bmatrix},$$

there is a total of  $N$  time sample points, and the dimension of the input matrix is  $N \times 3$ . Prediction task  $v_m(t)$  could be represented as  $Y$ . This is known as the high-dimensional sequence learning problem. A deep learning architecture with temporal processing capabilities is desired.

**2.2. Characteristics of Traffic Flow.** As a first step in the model building process, we made a preliminary investigation of the traffic patterns. We chose a common road network structure in Chinese cities and observed the traffic speed values in 180 minutes repeatedly. Figures 2-3 present the observed traffic speed values of relevant road sections under normal, incident, conditions, respectively. As shown in Figure 2, the trend of their change is relatively consistent and stable, waving up and down in small amplitude. From Figure 3, when there is a traffic incident, the traffic speed values of the road decline significantly. At the same time, the traffic speed values of upstream rise and the values of downstream decline slowly. The result shows that the traffic speed values will change significantly when the traffic incident happened, and its relationship with other road sections is significantly different compared to one in normal traffic state. So in the process of deep learning, there will be a worse training accuracy and the training time will be longer without distinguishing

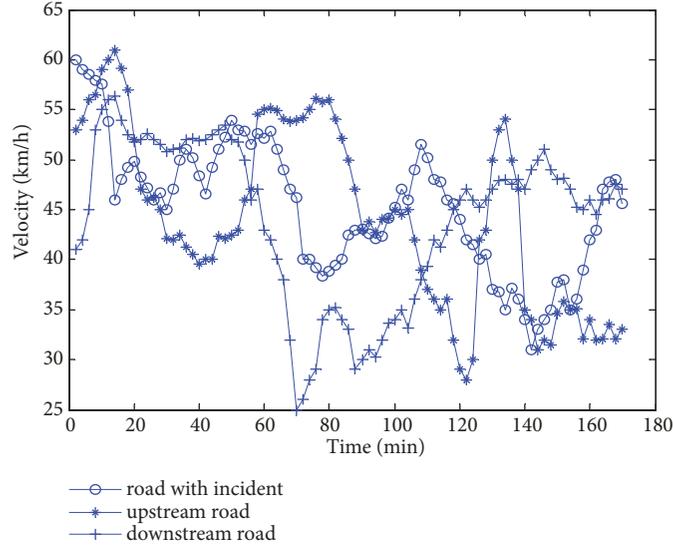


FIGURE 2: The traffic speed values of the relevant road sections in general traffic state.

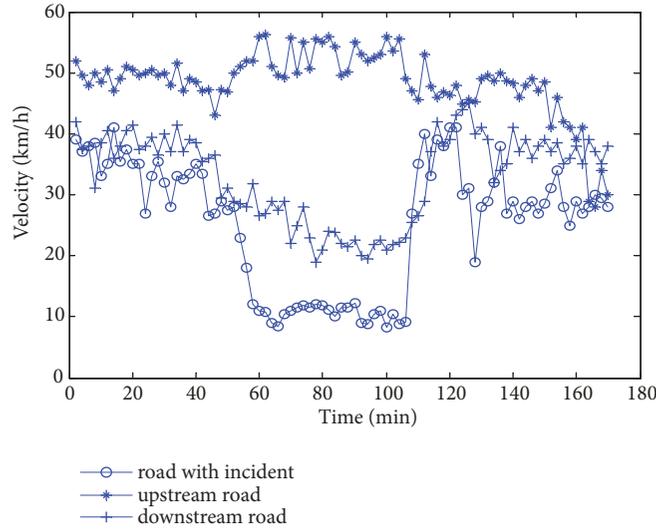


FIGURE 3: The traffic speed values of the relevant road sections in traffic incident.

between two kinds of traffic conditions. Therefore, a model with distinguishing the traffic state is proposed.

**2.3. Model Architecture.** The proposed classified deep neural network structure (CDNNs) is as shown in Figure 4. Since the spatial difference between normal and incident conditions is very large, only using one neural network cannot capture the dynamics effectively. So we classify the traffic into two clusters and train a neural network for each cluster. The pretraining module has two functions. One of them is to extract features of the input data for traffic classification through classification module. The other is to initialize the weights of architecture in fine-tuning module for traffic prediction.

Let the input dataset of the prediction model be denoted as  $X$ , and prediction task could be represented as  $Y$ .

**2.3.1. Pretraining Module.** Pretraining module serves to provide a deep learning through DBN to achieve the automatic extraction of abstract representations only using unlabeled data. We take the training data  $X$  as the input of DBN.

DBN is a stack of Restricted Boltzmann Machines which are energy-based models. Therefore, an energy function for the model is defined firstly [12]. At the same time, the real values with Gaussian noise are used to represent traffic data. Then, for a given set of states  $(v, h)$ , the energy functions and conditional probability distributions can be defined as follows:

$$\begin{aligned}
 -\log P(v, h) &\propto E(v, h; \theta) \\
 &= \sum_{i=1}^{|V|} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{|H|} b_j h_j
 \end{aligned}$$

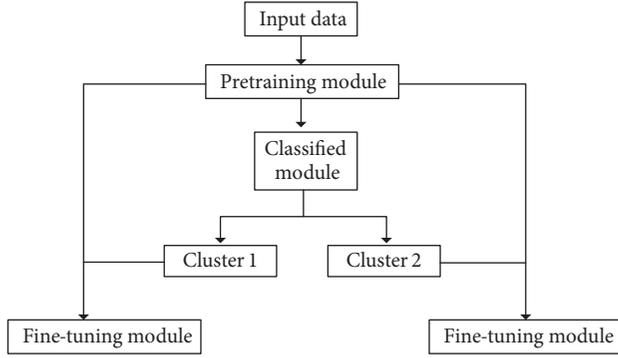


FIGURE 4: The proposed deep architecture.

$$-\sum_{i=1}^{|V|} \sum_{j=1}^{|H|} \frac{v_i}{\sigma_i} h_j w_{ij} \quad (1)$$

where  $v$  is visible unit,  $h$  is hidden unit,  $\theta = (w, a, b)$  are parameters of RBM, and  $w_{ij}$  is the connection weight of unit  $i$  and  $j$ , with  $a_i$  and  $b_j$  as their bias. Based on this energy function, we can get the conditional probability distributions of the state  $(v, h)$ :

$$\rho(h_j | v; \theta) = \text{sigm} \left( \sum_{i=1}^{|V|} w_{ij} v_i + b_j \right) \quad (2)$$

$$\rho(v_i | h; \theta) = N \left( a_i + \sigma_i \sum_{j=1}^{|H|} h_j w_{ij}, \sigma_i^2 \right) \quad (3)$$

where the Gaussian distribution is denoted as  $N(\mu, \sigma^2)$ , and  $\text{sigm}(x) = 1/(1 + e^{-x})$  is a sigmoid function. Then we can get the distribution of  $v$ :

$$p(v) = \sum_h p(h | \theta) p(v | h, \theta) \quad (4)$$

The process of training DBN is carried out layer by layer. In each layer, use the visible data vector to infer the hidden layer, and then this hidden layer is taken as the visible data vector of the next layer (higher layer) [13]. The parameters  $\theta = (w, a, b)$  are learned through CD algorithm [14].

DBN is thus used to learn unsupervised features.

**2.3.2. Classified Module.** The characteristics learned from DBN are the most representative for data. We can use DBN as the features learning model to classify data. So we just add a final classifier and fine-tune the model with the labeled data. The category labels are respectively denoted as normal traffic condition and traffic incident condition. The data are labeled according to the traffic information which was collected from traffic management department.

In this section, we use our learned features as direct input to the logistic regression which is suited for binary classification. Suppose there is  $n$  training samples  $X = \{X_1, \dots, X_n\}$ , where  $X_i$  is a vector of  $d$  dimension.  $Y = \{y_1, y_2\}$  are the

category labels: normal traffic condition and traffic incident condition; the logistic regression will study such a function:

$$f(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \quad (5)$$

Let

$$P(y = 1 | x, \theta) = f(x) \quad (6)$$

Then the probability belongs to the category of 0 is

$$P(y = 0 | x, \theta) = 1 - f(x) \quad (7)$$

We use the maximum likelihood estimation method to solve the parameters.

Then the input dataset  $X$  is classified into two subsets  $X_1, X_2$ .

**2.3.3. Fine-Tuning Module.** Previous studies have shown that differences of early parameters produce major influence on the final choice during training procedure. Therefore, the parameter being limited to a range through pretraining can achieve a more efficient optimization. A very good feature of a DBN is that it can infer the states of the layers of hidden units only in forward pass. Then the inference can be used in deriving the variational bound [15]. So after pretraining through a stack of RBMs, we can jettison the whole probabilistic framework and simply use the generative weights in the reverse direction as a way of initializing the weights of all except the last layer of a traditional deep neural network. Then, the ‘‘recognition’’ weights of the DBN become the weights of a standard neural network for each traffic condition respectively. In Section 2.3.2, the training data have been classified into two clusters according to the traffic condition. We then just add a final layer of variables that represent the desired outputs above each network and respectively fine-tune the model with each labeled data using backpropagation.

In conclusion, with pretraining completed in the first section, we add a logistic regression layer above the DBN to form a classifier. And the classifier is trained using the labeled (normal or incident) data. Then we train two deep neural networks. Thus, we can apply the complete module to predict the traffic flow. Firstly, the raw datasets without any additional information as the input of DBN are preprocessed, and they are classified into two clusters through the classifier. Then each group of the two classified datasets is used to forecast the traffic through the corresponding deep neural networks. An illustration of the deep architecture with classifier for traffic flow prediction is as shown in Figure 5. The raw data as input of DBN are preprocessed and the initialized weights are generated through training DBN layer by layer except the final layer of DBN. Next, the generated weights are firstly classified into two sets through adding the logistic regression as the final layer of DBN. Then the classified weights are used once again to predict the traffic flow through adding a final layer of variables that represent the desired outputs.

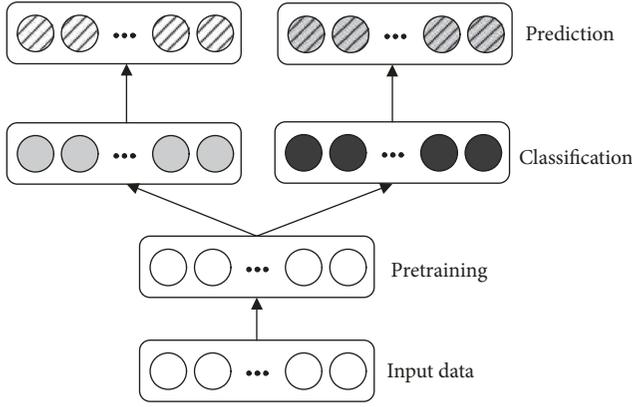


FIGURE 5: Illustration of the deep architecture with classifier.

### 3. Experiments and Results

#### 3.1. Experiment Settings

**Data Description.** Select three road segments randomly throughout Nanming District of Guiyang in Guizhou province as the research object. In Figure 6, the basic network of Nanming District is as shown, and the marked locations are the randomly selected road segments to collect data.

The spatial location and information of the road segments are as shown in Table 1. For each of the 3 selected segments, the traffic speed (vehicle average speed of each segment) data from 2011-04-11 and 2011-05-10 month-long were collected by coil detector and microwave detector. These 30-s raw data are then aggregated into 5-min periods. Besides detector data, the incident data which were recorded after incident were also collected from traffic management department. Then three time units of traffic data are taken as the input of the neural network, and the next time unit is taken as the future traffic data to make predictions. Stating in simpler words, we take 7:15, 7:20, and 7:25 time points as input units, including 20 raw data points from 7:15 to 7:25, to predict the traffic data at 7:30. We further aggregate the data into 15-minute periods as a set of training samples. Then we get 3 datasets with 86400 groups of sample data for each road segments, the first 20 days of data as the training data, the last 10 days of data as the test data. Therefore the original data matrix dimension is  $57600 \times 3$ ,  $28800 \times 3$ , respectively. And we repair the lost data and error data and make standardization of the data by normalizing the traffic speed into  $[0, 1]$ .

**Evaluation Metrics.** We use the absolute percent error (APE), mean absolute percent error (MAPE), and root mean square error (RMSE) for error measurement. They are defined as

$$APE (\%) = \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100 \quad (8)$$

$$MAPE (\%) = \frac{1}{n} \cdot \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100 \quad (9)$$

$$RMSE = \left[ \frac{1}{n} \sum_{i=1}^n (|y_i - \hat{y}_i|)^2 \right]^{1/2} \quad (10)$$

where  $\hat{y}_i$  represents the predicted traffic flow and  $y_i$  represents the observed traffic flow. We could get the mean accuracy (MA) to evaluate the prediction performance of each algorithm. The MA is employed as follows:

$$MA = 1 - \frac{1}{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (11)$$

**Selection of Parameters.** The network parameters of the deep architecture are chosen through numerous simulations on the training sets. The number of units in each layer is chosen from 40, 80, 160, 320, 640, 1280, and 2560. The layer size is set from 1 to 4. And the epoch is set from 10 to 100 with interval of 10. We tried all the parameters combinations 10 times for each test and chose the setting number of layers=3, number of units=160, epochs=40. In order to test the effect of each parameter on our deep architecture, we changed the value of one of the parameters while keeping the other parameters fixed. The result of network size is reported in Tables 2 and 3. Taking the MA and the training time into account, 3 layers and 160 nodes in each layer are the best choice.

Then, we investigate the effect of epochs. Figure 7 shows that the curve of accuracy on the training set is a function of the number of epochs. Apparently, large epoch, which would lead to a large temporal cost, is not appropriate in our model, although they can improve the accuracy on the training set.

For comparison purposes, three existing models are employed here. They are a neurofuzzy C-means model (FCM) [16], a deep learning architecture (DLA) [17], and the neural networks (NN) model [18]. The parameters of these architectures were trained by grid search using the same traffic datasets as the one used for the proposed CDNNs. Among these three methods, the FCM method is a hybrid model that combines neural networks with fuzzy C-means. The fuzzy C-means classifies traffic flow patterns into a couple of clusters. Then, the FCM model could forecast the traffic flow associated with each cluster. Through iterative experiments, the network architecture of FCM with 3 input nodes, 20 centers, was selected. The DLA method, which could learn effective features for traffic flow prediction, has a deep belief network at the bottom and a regression layer at the top. The architecture of DLA was composed of 3 layers, 128 nodes in layers and 40 epochs. The NN method is a backpropagation neural network with one hidden layer. The deep architecture of NN method is more responsive to dynamic conditions for the traffic flow forecast than the historical, data-based algorithm. After trial and error, the optimized architecture of NN was composed as follows: ten neurons in the input layer, single hidden layer with 4 neurons and 1 output neuron. The input data for the methods are the same with that for CDNNs model and all of the experiments were implemented in Matlab.

**3.2. Prediction Performance Results.** Firstly, the prediction performance of the proposed model (CDNNs) for target

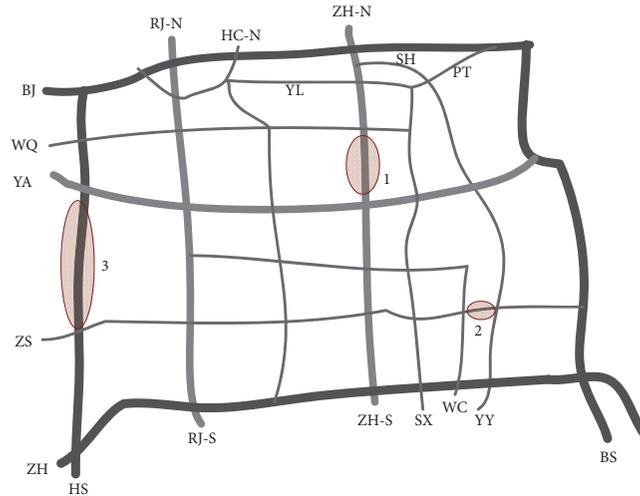


FIGURE 6: Basic network of Nanming District and data collection.

TABLE 1: The spatial location and information of the road segments.

Number	Target	Downstream	Upstream
1	WQ-YA-(ZH-N)	YL-WQ-(ZH-N)	YA-WC-(ZH-N)
2	WC-YY-(ZS)	SX-WC-(ZS)	YY-BS-(ZS)
3	YA-ZS-(HS)	WQ-YA-(HS)	ZS-ZH-(HS)

TABLE 2: Effect of the number of layers.

Layers	1	2	3	4	5	6	7
MA	0.855	0.886	0.901	0.912	0.915	0.921	0.926
Time	82s	209s	310s	455s	550s	661s	798s

TABLE 3: Effect of the nodes in a layer.

Nodes	40	80	160	320	640	1280	2560
MA	0.857	0.879	0.895	0.918	0.926	0.931	0.953
Time	72s	169s	412s	1455s	3150s	8661s	19923s

road WQ-YA-(ZH-N) on a randomly selected day is given in Figure 8, and Figure 8(a) presents the predicted values, and the APE values for the prediction are plotted in Figure 8(b). As shown in Figure 8(a), a traffic incident occurred at 18:50, blocking traffic until 19:30, which naturally dropped the traffic flow. From Figure 8(a), we can see that the predicted result draws near the actual data for most time even during the period of incident. This is also shown in Figure 8(b), where the APE values are plotted.

In order to clearly show the potential of CDNNs to respond well to the incident, the MAPE values of CDNNs for WQ-YA-(ZH-N) road segment in different scenarios are given in Table 2 for further analysis. The ‘average’, which refers to the average prediction for the whole period between 13:00 and 20:00, indicates the overall performance of CDNNs. The ‘without incident’ refers to the average prediction for the period between 13:00 and 18:50 and the period between 19:30 and 20:00. ‘Incident only’ refers to the average prediction for a

period between 18:50 and 19:30, when the incident happened and this value indicates the ability of CDNNs to respond to incident. From Figure 6(b) we can see that only at the beginning of the occurrence of the incident, the absolute percent error (APE) is extremely high, and then it quickly falls back. So calculating the average prediction for the period of ‘incident only’ with the certain point excluded can assess its capability to recover its prediction accuracy for incident. As shown in Table 4, the overall performance for the whole period is close to the performance for the normal traffic period. It presents much worse prediction performance for the traffic with incident because the unexpected incident caused time to react. Once the model identified the incident, the prediction performance for the traffic with incident will be even better than that for normal traffic, with an average MAPE of 5.93%.

To test the effectiveness of preclassification in the deep architecture, we compare the performance of the proposed

TABLE 4: Prediction performance in different scenarios.

Scenario	Average	Without incident	Incident only	Point excluded
MAPE(%)	6.72	6.28	9.66	5.93

TABLE 5: Prediction performance in different scenarios.

Scenario	MAPE(%)			
	CDNNs	FCM	DLA	NN
Overall	6.89	11.52	9.73	13.43
Without incident	6.27	9.18	6.51	10.85
Incident only	9.06	16.01	21.39	22.77

TABLE 6: Prediction performance in different scenarios.

Road	MAPE(%)				RMSE			
	CDNNs	FCM	DLA	NN	CDNNs	FCM	DLA	NN
1	6.49	11.52	9.73	13.43	3.98	4.57	3.55	5.42
2	6.89	8.92	6.5	11.02	3.48	5.72	4.89	6.36
3	6.69	8.59	6.85	11.17	3.99	5.08	4.05	6.31

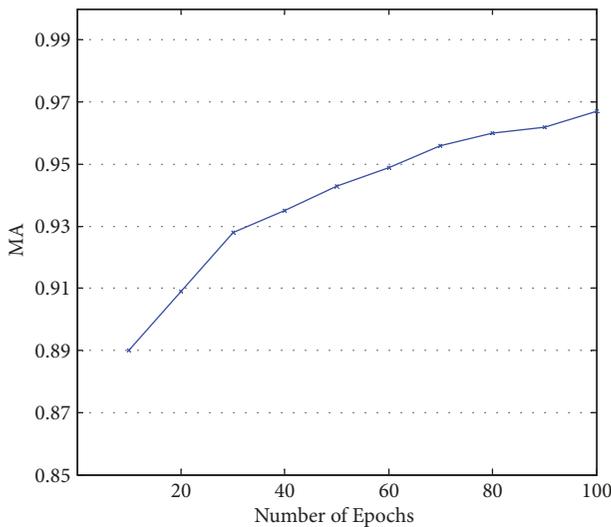


FIGURE 7: Effect of epochs.

CDNNs model with the other models which are introduced in Section 3.1 for road segments: WQ-YA-(ZH-N), WC-YY-(ZS), and YA-ZS-(HS), respectively.

Figure 9 shows prediction performance on a certain day (4th of May, which is chosen randomly) for WQ-YA-(ZH-N)(traffic incident period is between the two lines). The actual and forecasted values are presented in Figure 9(a). As expected, the CDNNs model has the best prediction performance. The APE values of each model are statistically not significantly different for most time, with the exception of incident period, when APE values are relatively higher, as shown in Figure 9(b). It can also be noticed in Figure 9(b) that CDNNs and DLA models present lower APE values during the normal traffic time periods. During the incident period, CDNNs clearly performs better than the other models,

because the incident traffic is equally treated as the normal one by the three models, and they could not respond to the unexpected changes well.

The MAPE of each model is computed by averaging the APE over three periods (the whole test time, the normal traffic time, and the incident time) singly, as shown in Table 5. It can be seen that CDNNs presents the best prediction performance for all three periods, with MAPE of 6.89%, 6.27%, and 9.06%, respectively. Under the incident conditions, CDNNs model has a much better prediction with MAPE of 9.06%, because it distinguishes traffic incident condition from the normal one and can effectively discover the spatial and temporal correlations between up-down streams even during the incident period. For normal traffic, DLA gives the second best performance, with MAPE of 6.51%, close to that of CDNNs, because it makes traffic flow prediction with deep learning similarly with CDNNs but without preprocess step. The performance of FCM for different periods is relatively stable compared with DLA and NN, because FCM categorizes traffic flow patterns into a couple of clusters. But it is not as sensitive as CDNNs due to lack of pretraining. Not surprisingly, NN model gives the lowest performance among all models, with MAPE of 13.43%, 10.85%, and 22.77%, respectively; this is due to the pure deep architecture of NN.

For road segments WC-YY-(ZS) and YA-ZS-(HS), where the models were tested on the same day as WQ-YA-(ZH-N), the prediction performances are presented in Figure 10, respectively. The results are coincident with the aforementioned conclusion.

In Table 6, the MAPE values and the root mean square error (RMSE) of each model for each road segments are given. It is clear that the predictions of CDNNs model outperform those obtained from the existing models. CDNNs model yields MAPE of around 6.5% compared to MAPE values from 6.5% to 11.52% obtained from the other models. The errors

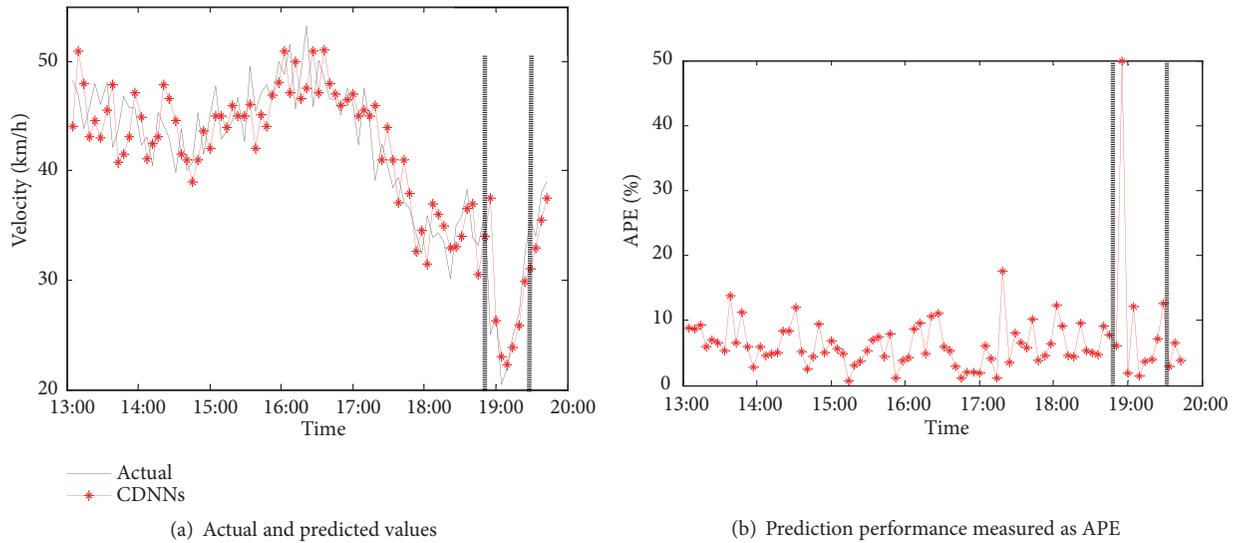


FIGURE 8: Performance for traffic flow prediction.

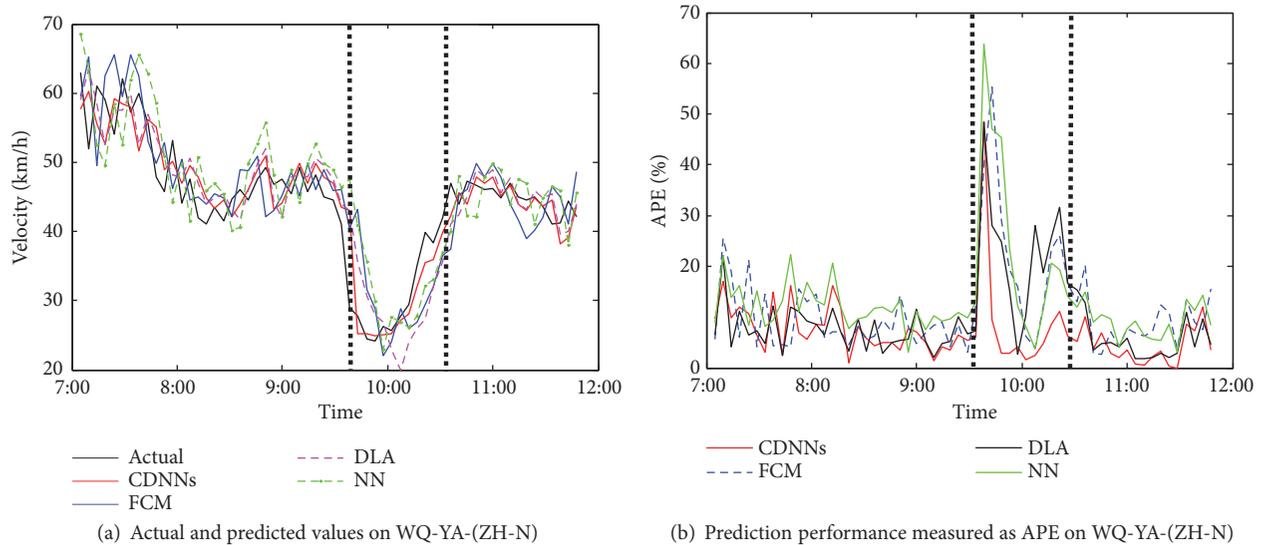


FIGURE 9: Performance for traffic flow prediction on WQ-YA-(ZH-N).

of CDNNs model have the lowest degree of being discrete with the RMSE values from 3.48 to 3.99 for the three road segments. It can be seen that the proposed model produces better prediction stability as compared to the compared models.

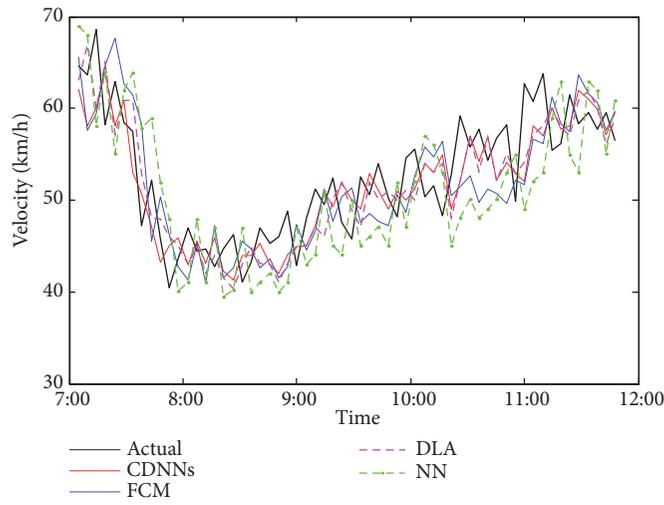
Finally, to see the overall prediction performance of 10-day test data, the prediction accuracy is computed by averaging the mean accuracy (MA) of 3 road segments over the 10 days. A visual display of the accuracy of the four models for both normal traffic and traffic with incidents on target road segments is as shown in Figure 11.

From Figure 11, we can see that the proposed CDNNs model could outperform the other three approaches, especially for the traffic with incidents. It consistently demonstrates the deduction described in previous section. We can

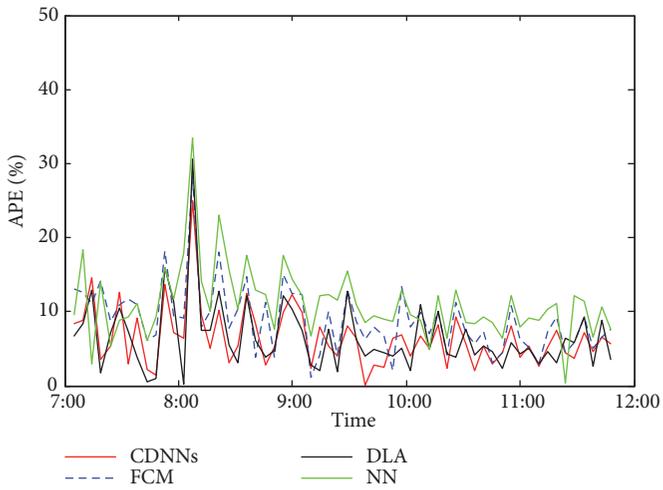
include that CDNNs model has a certain potential in complex nonlinear short-time traffic flow prediction.

#### 4. Conclusion

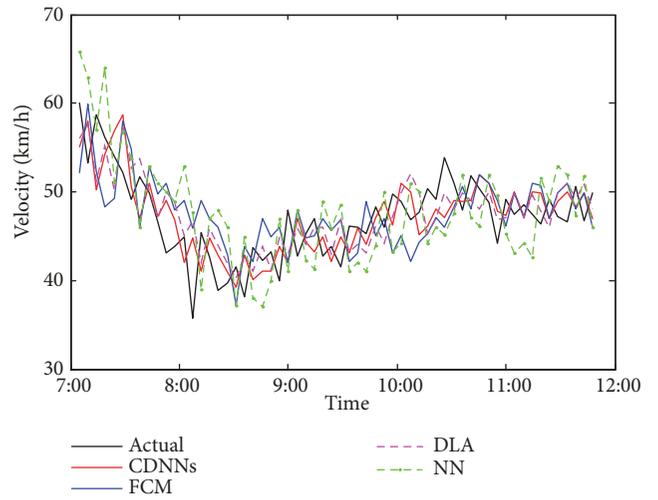
In this paper, a deep machine learning architecture consisting of three modules: a pretraining module, a classification module, and a fine-tuning module, has been proposed to predict the short-term traffic flow. The pretraining module took a stack of RBMs at the bottom, which is effective for unsupervised features learning. The classification module and the fine-tuning module put regression layers at the top to classify the traffic data into two traffic states and predict the traffic flow respectively. The performance of the proposed architecture clearly indicated the effectiveness in traffic flow



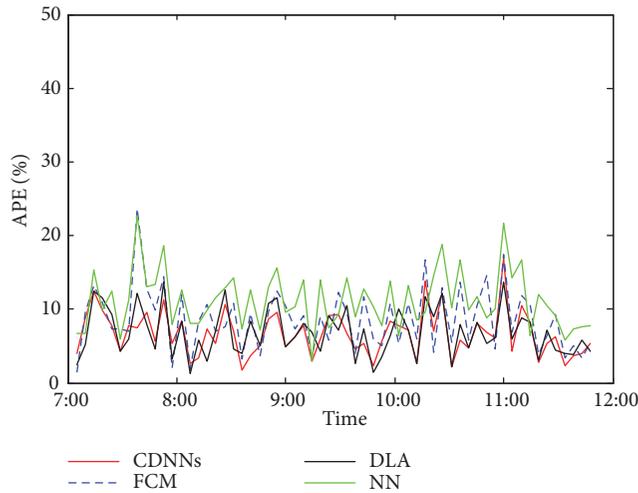
(a) Actual and predicted values on WC-YY-(ZS)



(b) Prediction performance measured as APE on WC-YY-(ZS)



(c) Actual and predicted values on YA-ZS-(HS)



(d) Prediction performance measured as APE on YA-ZS-(HS)

FIGURE 10: Performance for traffic flow prediction.

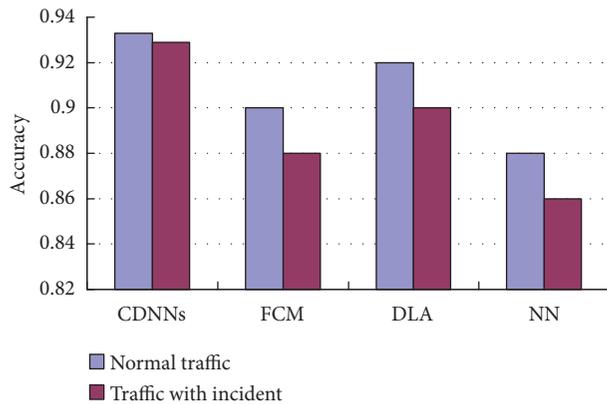


FIGURE 11: Comparison on prediction accuracy of four models.

prediction compared with the well-established FCM, DLA, and NN models. Distinguishing the incident from traffic flow contributes to optimizing the weights and improving the performance of the features learning. However, employing classification module increases the computational time though speeding up the running time of the fine-tune module. In future work, we will investigate ways to reduce the training time of our models, such as sparse representation of data and integration of the classification module with fine-tuning module.

## Data Availability

The data used to support the findings of this study are available in the supplementary materials file (available here).

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the Natural Science Foundation of Hebei Province [G2014203219] and National Natural Science Foundation of China [71171174].

## Supplementary Materials

Table 1. The spatial location and information of the road segments. (*Supplementary Materials*)

## References

- [1] M. van der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.
- [2] C. K. Moorthy and B. G. Ratcliffe, "Short term traffic forecasting using time series methods," *Transportation Planning and Technology*, vol. 12, no. 1, pp. 45–56, 1988.
- [3] J. Ma, X.-D. Li, and Y. Meng, "Research of urban traffic flow forecasting based on neural network," *Acta Electronica Sinica*, vol. 37, no. 5, pp. 1092–1094, 2009.
- [4] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction: neural network approach," *Transportation Research Record*, vol. 1453, 1994.
- [5] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.
- [6] S. Clark, "Traffic prediction using multivariate nonparametric regression," *Journal of Transportation Engineering*, vol. 129, no. 2, pp. 161–168, 2003.
- [7] 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, pp. 1–17, 2015.
- [8] D. Srinivasan, C. Wai Chan, and P. G. Balaji, "Computational intelligence-based congestion prediction for a dynamic urban street network," *Neurocomputing*, vol. 72, no. 10–12, pp. 2710–2716, 2009.
- [9] H. B. Yin, S. C. Wong, J. Xu, and C. K. Wong, "Urban traffic flow prediction using a fuzzy-neural approach," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 2, pp. 85–98, 2002.
- [10] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 2, pp. 121–135, 2003.
- [11] B. Pan, U. Demiryurek, and C. Shahabi, "Utilizing real-world transportation data for accurate traffic prediction," in *Proceedings of the 12th IEEE International Conference on Data Mining, ICDM 2012*, pp. 595–604, Belgium, December 2012.
- [12] Y. W. Teh and G. E. Hinton, "Rate-coded restricted boltzmann machines for face recognition," in *Proceedings of the 14th Annual Neural Information Processing Systems Conference, NIPS 2000*, pp. 908–914, USA, December 2000.
- [13] A.-R. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.
- [14] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [15] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [16] B. Park, "Hybrid neuro-fuzzy application in short-term freeway traffic volume forecasting," *Transportation Research Record*, no. 1802, pp. 190–196, 2002.
- [17] W. Huang, H. Hong, M. Li, W. Hu, G. Song, and K. Xie, "Deep architecture for traffic flow prediction," in *Advanced Data Mining and Applications*, vol. 8347 of *Lecture Notes in Computer Science*, pp. 165–176, Springer, Berlin, Germany, 2013.
- [18] B. L. Smith and M. J. Demetsky, "Short-term traffic flow prediction: neural network approach," *Transportation Research Record*, vol. 1453, pp. 98–104, 1994.

