Received August 1, 2018, accepted August 26, 2018, date of publication September 4, 2018, date of current version October 8, 2018.
Digital Object Identifier 10.1109/ACCESS.2018.2868735

Research on Traffic Speed Prediction by Temporal Clustering Analysis and Convolutional Neural Network With Deformable Kernels (May, 2018)

GUOJIANG SHEN[1], CHAOHUAN CHEN[1], QIHONG PAN[2], SI SHEN[1], AND ZHI LIU[1]
[1]Computer Intelligence System Institute, Zhejiang University of Technology, Hangzhou 310023, China
[2]Electrical and Computer Department, Colorado State University, Fort Collins, CO 80521, USA
Corresponding author: Zhi Liu (lzhi@zjut.edu.cn)

ABSTRACT Real-time traffic speed prediction is one of the most essential parts of the Intelligent Transportation System. In recent years, with the development of artificial intelligence technology, such as in-depth learning, new prediction methods have emerged in endlessly and achieved good results. However, the spatio-temporal information, traffic environment, and their interaction have been hardly depicted in these methods. Therefore, a novel traffic speed prediction method based on temporal clustering analysis and deformable convolution neural network (TCA-DCNN) is proposed in this paper. Temporal clustering analysis (TCA) based on Differential Evolution and hierarchical clustering can adaptively distinguish traffic environment by discriminating the traffic speed variation pattern. By further introducing the deformable convolutional kernels, the characteristics of spatio-temporal traffic speed variation are precisely located in deformable perception fields. After that, a set of DCNN models are trained by using the data processed by TCA, and the output of one of the basic DCNN model is selected as the final traffic speed prediction result. The simulation results based on the measured data in Hangzhou, China, show that the TCA-DCNN algorithm has better performance than other algorithms in traffic speed prediction.

INDEX TERMS Intelligent transportation system, traffic speed prediction, deformable convolution neural network, temporal clustering analysis.

I. INTRODUCTION

Real-time traffic speed prediction in urban road network remains a vital and challenging topic for transportation researcher, especially in advanced traffic navigation systems and traffic management systems. To settle this issue, a great deal of prediction models have been proposed. Most of them either relied on model-driven approaches criticized for strong assumptions and inevitable bias or employed data-driven approaches having become increasingly popular for the booming machine learning and artificial intelligence algorithms. With the rapid development of big data and artificial intelligence technology, deep learning is considered to be one of the state-of-the-art prediction approaches. Due to the powerful nonlinear fitting capability and high-order correlation architecture of deep model, there is significant growth in spatio-temporal range of training data. And other related data, such as weather information and social media data, can easily be gathered into the prediction model, without worrying about curse of dimension.

Among the mainstream deep architectures, convolutional neural network (CNN) is famous for its outstanding image-processing capability. Its main contribution is to use convolution kernel to extract local features through sliding windows. If traffic speed data are expanded according to space axis and time axis, spatial relationship between upstream and downstream and temporal correlation of continuous time steps can be clearly presented. Strong correlation mostly exists between samples with small Euclidean distance, which is coincident with a 2D image. Therefore, CNN is considered to have strong ability to predict traffic speed. Some studies [1], [2] have proven CNN's excellent data mining capability, especially for spatio-temporal correlation of traffic speed data. However, the disadvantages of CNN are also obvious: the convolutional kernels sample the input feature on fixed grid, and the pooling layers reduce the spatial resolution at a fixed proportion [1]. These means that CNNs cannot be adjusted adaptively with respect to training data's spatial features. To settle this issue, deformable convolutional networks (DCNN) is introduced [1], [2]. Just as its name imply, with additional offsets learning from the target tasks, DCNN's convolution kernels are deformable, which makes the obtaining of deformable features feasible. In this article, a DCNN traffic speed forecasting model is proposed, in which spatial and temporal traffic speed variation features are precisely located in deformable perception fields, which can lead to critical improvement on prediction performance.

In addition, previous studies [3], [4] have shown that the predictor considering traffic environment may be better than others. However, these studies usually convert environment to certain value by historical experience and input it into the model [3]. In this paper, an effective approach is proposed which builds a set of basic predictors suited different environments, and choose an optimal predictor based on the actual environment for a more effective prediction result. Before that, traffic environment should be partitioned first. Many features can be used to distinguish a traffic environment, such as time of day, weather, temperature, location, number of lanes etc. However, there are many types of features of traffic environment and some types of features are difficult to obtain. Due to the different patterns of traffic speed variation correspond to its traffic environment, traffic environment can be finely partitioned by distinguishing traffic speed variation patterns. Before that, the data reconstruction algorithm[5]–[7] should be used to extract the traffic speed from the data of passing vehicles.

The novel contributions of our study are summarized as follows.

(1) A method of environment partitioning based on DE and HC is presented. By this method, the velocity data and its corresponding environment can be accurately divided into several parts.

(2) A method of extracting spatio-temporal variation features of traffic speed based on DCNN is proposed. The deformable convolution layer is introduced into DCNN to change the size of the kernels and enhance the ability of DCNN to model the variation of spatio-temporal information caused by the change of environment.

(3) An optimal prediction model selection method is designed. A set of basic DCNN models suited different environments are constructed and an optimal model is selected as the final predictor from basic DCNN models according to the sample mean rewards.

The remainder of the paper is organized as follows. Section II reviews the related work. Section III introduces the DCNN which used in section IV. Prediction model is depicted in section IV, and experimental results with real world traffic datasets are reported in section V. At the end of this paper, the conclusions are presented and future studies are discussed.

II. RELATED WORK

In recent years, a large number of articles on traffic parameter prediction have been published. The prediction models mainly include two kinds: parametric models and non-parametric models. Among the parametric models, a wildly used method is the autoregressive integrated moving average model (ARIMA), which is a time-series prediction model and takes the correlation of continuous time-series of traffic characteristic data into account. Since the birth of the Box-Jenkins time-series analyzing method [8], many ARIMA-based variants have been proposed, such as Seasonal ARIMA [4], KARIMA [9] and ARIMAX [10]. Another commonly used parameters model is Kalman filtering, which is a time-domain state-space method and, has the advantage of generating prediction models with a small amount of data for ARIMA. Some researchers applied Kalman filtering to forecast traffic flow [11] and travel time [12].

As nonparametric models used in traffic forecasting, great attention has been paid to Support Vector Regression (SVR) [13] and the Artificial Neural Network (ANN) [14]. SVR models have relatively strongly global capability to find the optimal solution. Research on SVR prediction model mainly lies in parameter selection, which greatly affect the final results. Thus, many researchers designed various algorithms to optimize the parameter selection of SVR. The improved models include the GA-SVR model which optimizes the SVR parameter selection by GA genetic algorithm [15], [16], the PSO-SVRR model optimized by PSO [17], and the chaotic PSO Swarm Optimization (CPSO-SVR) algorithm [18], etc.. ANN is an abstraction of human brain neuron network, and its main advantages are strong nonlinear fitting capability and great generalization ability. With the explosion of deep learning technology, a large number of different neural network models are emerging. ITS Researchers also began to use the neural network model as effective traffic speed predictor. Lv *et al.* [19] proposed a deep architecture model applied with autoencoders as building blocks to represent traffic flow prediction. Ma *et al.* [20] used RNN-RBM model to predict the evolution of large-scale congestion in a traffic network. SAE models are adopted to predict traffic characteristic parameters [21]. Some models also consider the effect of rainfall on traffic flow [3]. Some methods combine the spatio-temporal information of traffic speed with the traffic flow predicted by long short-Term memory (LSTM) network and CNN [2], [22]. Wang et al. [1] proposed an error Convolutional Neural Network (eCNN) model that combines CNN and RNN (Recurrent Neural Network) to reduce the error in traffic speed prediction especially in traffic events with speed dramatic variation. Generally speaking, it is widely believed that deep neural networks are the state-of-the-art approaches for traffic speed prediction. However, most of the existing research results haven't make full use of its privileges in prediction

III. CONVOLUTION NEURAL NETWORK WITH DEFORMABLE KERNELS

In this section, we introduce the convolution neural network with deformable kernels which is used in our prediction models on section IV. Fig.1 shows the framework of DCNN

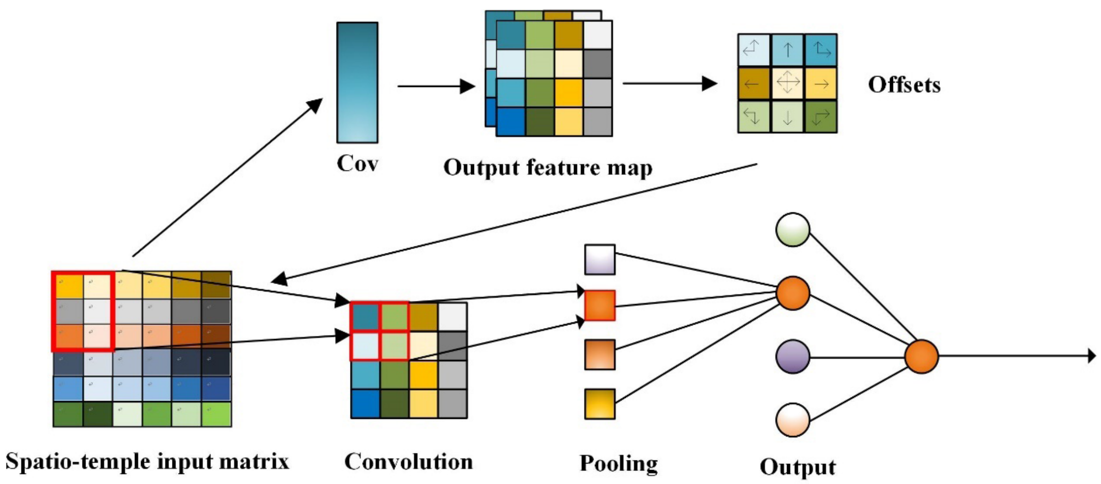


FIGURE 1. The framework of DCNN model

that contains four main layers: input layer, deformable convolutional layer, pooling layer and output layer. Input layer is traffic speed matrix of objective road section and its adjacent sections from previous intervals. Convolution layer is to extract the features of Spatio-temporal matrix by deformable convolution kernels. Pooling layer is used to reduce the feature dimension from convolutional layer. Output layer is to generate the prediction result of traffic speed.

A. THE INPUT LAYER

In order to utilize Spatio-temporal information, we design a spatio-temporal matrix as the input of input layer. Traffic speed of road r at time t can be defined as $V_{r,t}$. Suppose number of previous intervals is j, input matrix can be expressed as follows:

$$\mathbf{X} = \begin{bmatrix} v_{r-i,t} & v_{r-i,t-1} & v_{r-i,t-2} & \cdots & v_{r-i,t-j} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ v_{r-1,t} & v_{r-1,t-1} & v_{r-1,t-2} & \cdots & v_{r-1,t-j} \\ v_{r,t} & v_{r,t-1} & v_{r,t-2} & \cdots & v_{r,t-j} \\ v_{r+1,t} & v_{r+1,t-1} & v_{r+1,t-2} & \cdots & v_{r+1,t-j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{r+i,t} & v_{r+i,t-1} & v_{r+i,t-2} & \cdots & v_{r+i,t-j} \end{bmatrix}. \tag{1}$$

As in (1), the row vector contains traffic speed in the same road from continuous time intervals (from t to $t-j$), the columns contain traffic speed in all the road from $r-i$ to $r+i$ at the same time interval. Besides, road numbered less than r located in the upstream of road r and road numbered greater than r located in the downstream. Thus, the input matrix basically contains spatio-temporal information.

B. DEFORMABLE CONVOLUTION LAYER

Convolution layer is the most important part of convolutional neural network, which extracts feature map from input layer by convolution kernels. In order to learn spatio-temporal features in different traffic speed variation patterns, we adopt deformable convolution kernels to connect the spatio-temporal input matrix in convolution layer. As illustrated in Fig.1 and Fig.2, two steps are different with traditional convolution: (1) generating offsets, and (2) generating deformable kernel of convolution.

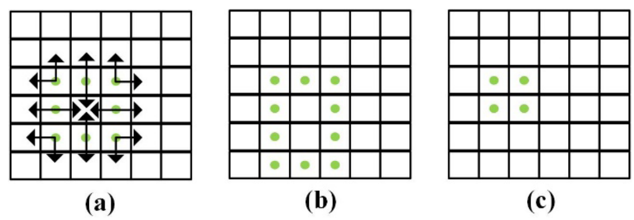


FIGURE 2. An illustrative example of the sampling locations in 3 ∗ 3 standard and deformable convolutions. (a) the deformed sampling locations (green points) with augmented offsets (black arrows) in deformable convolution. (b) and (c) the result of deformable convolution.

The offset filed comes from preceding feature maps via additional convolutional layer, whose channels are twice as many as the spatio-temporal input matrix. Besides, the deformation is conditioned on the spatio-temporal matrix in adaptive manner.

There is a formula to illustrate the process of generating deformable convolution:

$$E(k_0) = \sum_{k_n \in Z} w(k_n) \cdot V(k_0 + k_n + \Delta k_n). \tag{2}$$

In deformable convolution, the scanning area is augmented with offsets, which defined in formula (2) as $V(k_0 + k_n + \Delta k_n)$. V denotes the whole spatio-temporal input matrix. k_0 is scanning location. k_n enumerates the location in traditional 3∗3 standard kernel Z defined as $\mathbf{Z} = \{[-1,-1], [-1,0], \cdots [1,1]\}$. Δk_n refers to the offsets of kernel which learned from spatio-temporal input matrix by additional convolution kernel. The algorithm is called via

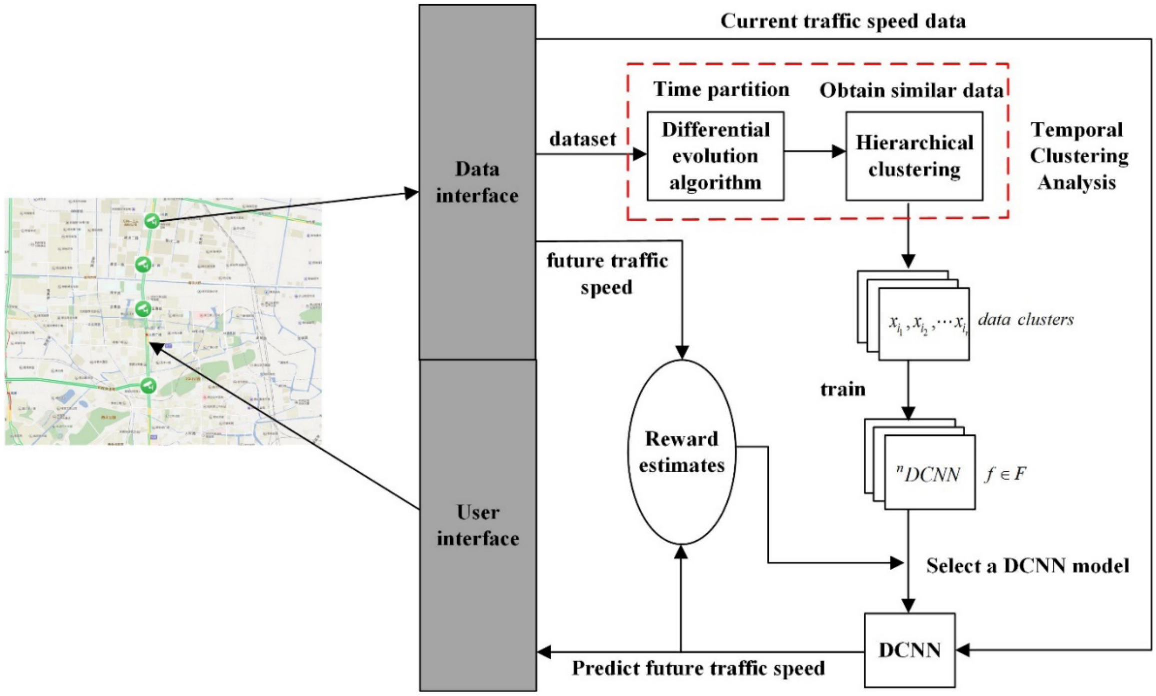


FIGURE 3. System diagram.

bilinear interpolation implement the offset value Δk_n as

$$X(k) = \sum\nolimits_{s} G(s, k) \cdot x(s). \tag{3}$$

The k of formula (3) refers to the kernel location, shape and size ($k = k_0 + k_n + k_n$), and k enumerates integral space location of all feature map. $G(\cdot;\cdot)$ is the kernel of via bilinear interpolation. Besides, G is a two-dimension kernel which can be split into two one-dimension kernels as

$$G(s, k) = g(s_x, k_x) \cdot g(s_v, k_v). \tag{4}$$

During train period, the convolution kernel generates feature map and learning offset. Besides, in order to learn offset, formula (4) uses via bilinear interpolation to make gradient back propagation. More details can be seen in [23].

C. POOLING LAYER

Pooling layer is another important layer, whose main role is to reduce the dimension of feature map from convolutional layer. We define the function as follow:

$$E_j^i = f(\sum \beta_j^i reduce(E_l^{i-1}) + b_j^i). \tag{5}$$

Function $reduce(\cdot)$ denotes down-sampling transformation, we generally use max pooling function or average pooling function. Down- sampling function transforms the feature map into smaller one. Then, the feature map output from Down-sampling function will multiply a multiplicative bias β and add a bias b.

D. OUTPUT LAYER

Output layer is similar to the traditional ANN, which is a fully connected layer and uses the feature map from pooling layer as the input. it generates the final predicted value as

$$y_j^i = \sigma(\sum (w_j^i E_i^{j-1} + b_j^i)) \tag{6}$$

where $\sigma(\cdot)$ is an activated function which usually is the RELU function; E_i^{j-1} denotes the input from the previous neuron. In order to get E_i^{j-1}, should be multiplied by a multiplicative bias w_i^i and add a bias b_i^i.

IV. METHODOLOGY

Fig.3 illustrates the system model under consideration. We use temporal clustering analysis based on differential evolution (DE) and hierarchical clustering (HC) to partition historical traffic data with its corresponding traffic environment, which can be used to train DCNN as set of basic predictors. After that, the most effective predictor is chosen as final predictor according to the estimation reward.

Based on the current traffic speed x_t from researched road, we aim to predict the future traffic speed y_t in the short term, eg., in next 5 minutes or 10 minutes. The t denotes the arrival time of the predicted request within a day. Each predicted request has traffic environment and spatio-temporal information, so the traffic environment should be distinguished to improve prediction accuracy. However, traffic environment is very complex and there is no clear boundary to partition traffic environment. Because different traffic environment leads to different traffic speed variations [24], we try

Algorithm 1 Temporal Clustering Analysis

Input: *dataset* Q, **population** M, **generation** T
Output: a set of basic predictors F

1 **for** D=2 to 10 **do:**
2   **Initialize** t=1.
3   **for** i=1 to M **do:**
4   **for** j=1 to D **do:**
5       $l_{i,t}^{j} = l_{i,t}^{j} + rand(0,1) \cdot (l_{\max}^{j} - l_{\min}^{j})$.
6   **end**
7   **end**
8   **while** $(|\partial(\Delta)| \geq \varepsilon \ or\ t \leq T)$ **do:**
9   **for** i=1 to M **do:**
10   **for** j =1 to D **do:**
11         $v_{i,t}^{j} = Mutation(l_{i,t}^{j})$.
12         $u_{i,t}^{j} = Crossover(l_{i,t}^{j}, v_{i,t}^{j})$.
13   **end**
14   **if** $\partial(\mathbf{u_{i,t}}) > \partial(\mathbf{l_{i,t}})$ **then:**
15         $\mathbf{l_{i,t}} = \mathbf{u_{i,t}}$.
16   **if** $\partial(\mathbf{l_{i,t}}) > \partial(\Delta)$ **then:**
17           $\Delta = \mathbf{l_{i,t}}$.
18   **end**
19   **else:**
20           $\mathbf{l_{i,t}} = \mathbf{l_{i,t}}$.
21   **end**
22   **end**
23   t=t+1.
24   **end**
25 **end**
26 $\mathbf{L} = \mathbf{l_{i,t}}$ *where* $\partial(\mathbf{l_{i,t}})$ *is* max.
27 $\mathbf{Q}' =$ partition$(\mathbf{Q}, \mathbf{L})$.
28 **for** each two $\mathbf{cluster_i}$, $\mathbf{cluster_j} \in$ *dataset* $\mathbf{Q}'$ **do:**
29   $similarity = \varphi(\mathbf{cluster_i}, \mathbf{cluster_j})$.
30   $\mathbf{S}.append(similarity)$.
31   $similarity_{\max}, i, j = find(\mathbf{S})$.
32   **if** $similarity_{\max} < threshold$ **do:**
33   go to line 35.
34   **else**
35     $\mathbf{cluster_k} =$ aggregate$(\mathbf{cluster_i}, \mathbf{cluster_j})$.
36   **end**
37 **end**

to partition traffic environment by analyzing traffic speed variation pattern from historical data. We use η and ω to denote the current traffic environment and traffic speed variation pattern respectively. The system contains a number of basic predictors $f \in \mathbf{F}$ where $\mathbf{F}$ is a predictor set. These basic predictors integrate the current speed x_t as spatio-temporal input matrix mentioned in section III and output the predicted speed x_t. These basic DCNN models are trained by historical data which contain different traffic speed variation patterns. However, we don't know which basic DCNN model suits the current traffic environment. We try to construct super predictor that choose the most effective basic DCNN model for current traffic environment. Thus, for each incoming request, the system chooses the output of one basic DCNN as the final prediction result y_t. Once the real traffic speed $\hat{y}_t$ in the next interval are received, the reward could be got by function $reward(y^t, \hat{y}^t)$ by comparing y^t and $\hat{y}^t$. $reward(\cdot, \cdot)$ reflects the magnitude of the error between the real value and the predicted value. As mentioned, each basic predictor is a traffic speed prediction model, whose effectiveness is estimated by its reward.

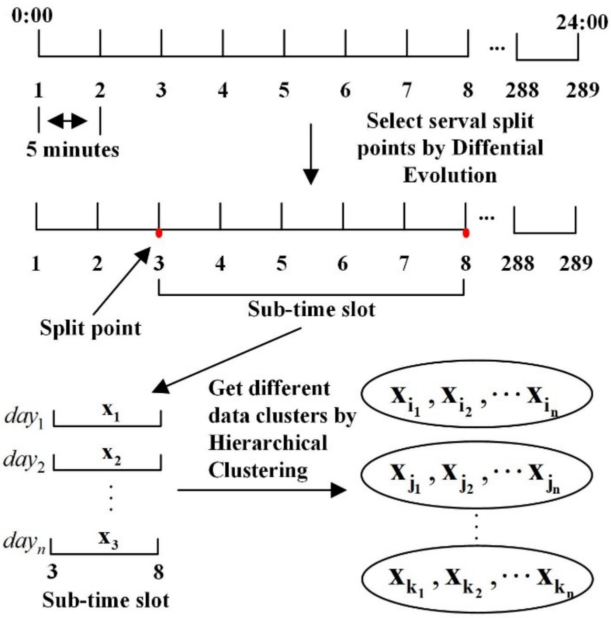


FIGURE 4. An illustrative example for temporal clustering analysis.

A. TEMPORAL CLUSTERING

In this subsection, we describe TCA algorithm. First of all, we introduce some useful concepts for describing the proposed algorithm.

- **Entire time.** The entire time T refers to a period of a day from 0:00 am to 24:00 pm. It consists of 288 basic time slots with 5 mins length, which partitioned by 289 continuous partition point.
- **Sub-time slot.** The sub-time slot t is created by partition T $t \in T$.
- **Partition point.** Partition point is one of 289 continuous points used to partition T into several t.
- **Time partition**. A time partition is a set of sub-time slot that cover entire time. for example, if split points are [9:00,15:00] and [10:00, 14:00, 18:00], {[0:00-9:00], [9:00-15:00],[15:00-24:00]} and {[0:00-10:00], [10:00-14:00],[14:00-18:00],[18:00-24:00]} are two different partitions.
- **Single data cluster**. Single data cluster is a dataset which contains traffic speed data in a sub-time slot of a day as $\mathbf{x_{t,d}} = [v_i, v_{i+1}, \cdots, v_{i+n}]$, the t and d denotes the sub-time slot and date respectively.

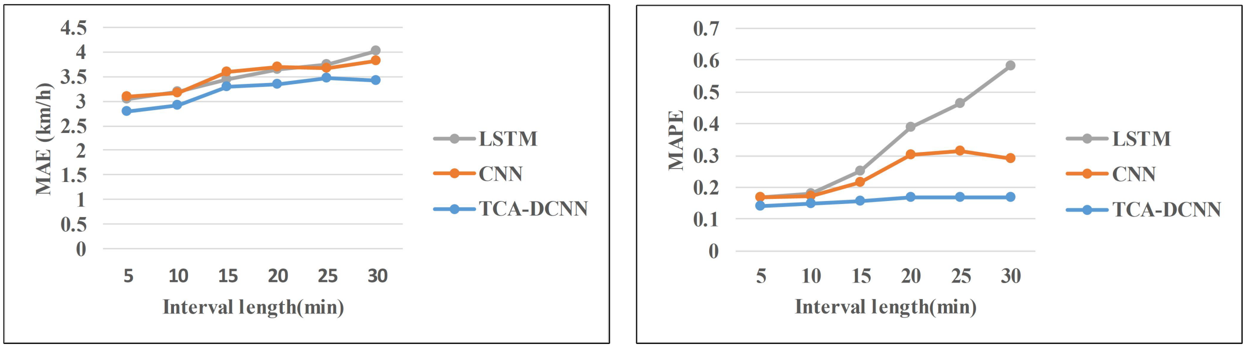


FIGURE 5. Segments of researched road.

- **Data clusters.** Data clusters consist of one or several single data clusters in same sub-time slot of different day. For example, $[\mathbf{x_{t_1,j_1}}, \mathbf{x_{t_1,j_2}}, \mathbf{x_{t_1,j_3}}]$ and $[\mathbf{x_{t_1,k_1}}, \mathbf{x_{t_1,k_2}}]$ are two different data clusters.
- **Similarity.** Similarity is a value that calculated by correlation coefficent of traffic speed in sub-time slot between two different data clusters. We use the Pearson correlation coefficient as the similarity measurement.

We will describe the algorithm in two main parts (see detail in formal description of algorithm 1). The first part (line1-line 26) is time partition based on differential evolution algorithm. This part deals with the population of Mindividuals encoded as $\mathbf{L} = [\mathbf{l_1}, \mathbf{l_2}, \cdots, \mathbf{l_M}]$. Each individual gene represents the location of each split point encoded as $\mathbf{l_{i,t}} = [l^1_{i,t}, l^2_{i,t}, l^3_{i,t} \cdots l^D_{i,t}]$. After random initialization, the population iteratively crossover and mutation to change the gene value heuristically, and the greedy algorithm based on adaptive function is used to select the individuals with higher fitness for the next generation. The adaptive function $\partial(\mathbf{I_{i,t}})$ calculates the average correlation coefficient among traffic speed data in each sub time slot of a day. Besides, the best number of split point D for the whole algorithm can be found by traversing D from 2 to 6. The second part (line27-line36) uses hierarchical clustering to obtain several data clusters containing many single data points. The data points have similar speed variation pattern for each of the same sub-time slot on different dates. Firstly, similarity matrix $\mathbf{S}$ is initialized. Then, the similarity between each two data clusters is calculated iteratively, and the two data with the largest similarity are clustered to the maximum similarity below the threshold. Note that the clustering used here is hierarchical clustering algorithm based on Pearson correlation coefficient.

Fig.4 provide an illustrative example of temporal clustering analysis in which the number of partition point is only 2. In this example, partition points are selected in 3 and 8 and entire time is partitioned to three parts. To predict traffic speed in period between split point 3 and 8, traffic speed data in the sub-time slot between partition point 3 and 8 of n different days need to be given. Finally, the similar patterns variation of traffic speed is found out by hierarchical clustering and is merged into a data cluster.

B. DCNN TRAINING

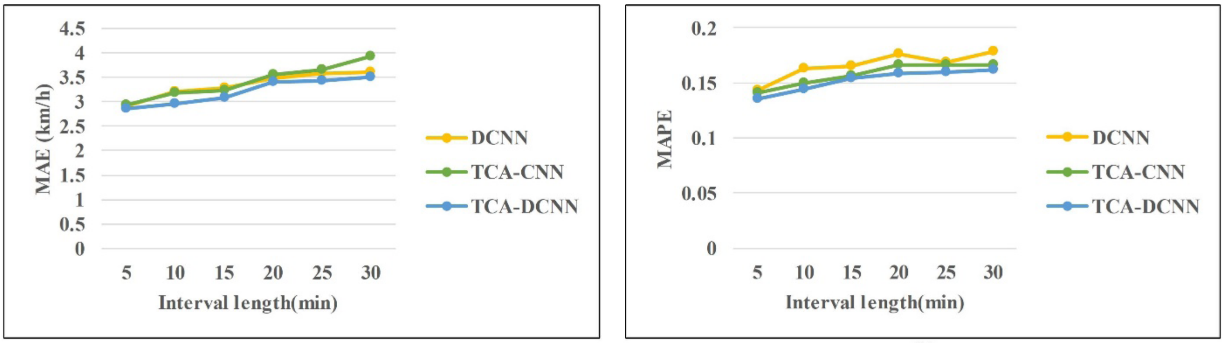
The parameters and kernel shape need to be trained in DCNN. The process is defined as

$$f = train(f', cluster_i), \quad f \in \mathbf{F}. \tag{7}$$

A set of basic predictors be trained by different data clusters from temporal clustering analysis. In (7), f' and f denotes the untrained DCNN model and trained DCNN model respectively, $cluster_i$ denotes each cluster data from temporal clustering. The training algorithm includes two aspects, one is training the offset of convolutional kernel and the other is training CNN. The two training processes are achieved by gradient descent method, and can be stated as follows: (1) Preparing data. Each data cluster from temporal clustering analysis is used as dataset for each untrained basic DCNN model. (2) Training offsets. Offsets are obtained from feature map by additional convolutional layer. In order to learn the offsets, the gradients are backpropagated by bilinear interpolation. (3) Training DCNN. Firstly, the parameters of DCNN and convolutional kernels is initialized respectively. Then, the backpropagated is adopt to train the parameter to minimize the loss function. Because all basic DCNN models are constructed from historical traffic speed data the performance of basic model is unknown before the real traffic speed varies with time. Therefore, the basic model should be retained after a cycle.

C. PREDICTOR SELECTION AND REWARD ESTIMATION UPDATE

This subsection describes the predictor selection and reward estimation updates (see detail in algorithm 2). When a traffic prediction request is coming, the current traffic speed vector of continuous segments are given. Then, all basic DCNN predictors, $\forall f \in F$ is active. Only one DCNN predictor of all basic DCNN predictors is chosen as the final predictor.



(a) (b)

FIGURE 6. Overall performance with varying interval lengths.

Algorithm 2 Predictor Selection and Reward Estimate Update

---

for every prediction request arrive (sub-time slot) **do:**
 Initialize the $reward_f(t) = 0, \forall f \in \mathbf{F}$.
 Generate the predictions result from all basic DCNN models $y^t = \tilde{f}(x^t)$ as (8).
 The true traffic speed $\hat{y}^t$ is revealed.
 Update the sample mean reward $reward_f(t)$.
end

The selection is as follows

$$y_t = \tilde{f}(x_t) \quad \text{where } \tilde{f} = \arg\max_f reward_f(t). \tag{8}$$

Over all, the selected basic DCNN predictor has the highest reward estimation at present sub-time slot among all models. This is a selection based on the sample mean reward in sub-time slot t. After the real traffic speed is given, the sample mean reward estimation for all basic DCNN models are updated.

V. EXPERIMENT

A. DATA DESCRIPTION

The simulation experiment of the proposed TCA-DCNN algorithm is carried out by use of the data collected by Hangzhou Traffic Police Brigade, Zhejiang, China. The traffic data of passing-vehicle collected from 8 observation stations in Xiaoshan District of Hangzhou are shown in figure.5. Each observation station has been equipped with cameras to capture traffic data on passing vehicles.

B. HANDLE THE DATA

The way of handling the data is similar to that described in paper [5]–[7]. In addition, the cameras of observation station have recorded every car that passing the intersection of the road. Firstly, traffic speed is calculated by matching the time information of all passing-vehicles through each observation point. Secondly, some work in data cleaning should be done because of some limitations of Bayonet. Some abnormal data should reset. for example, if some elements in spatio-temporal input matrix are larger than 60km/h or equal to 0 km/h with zero traffic flows, these data is set to be 25km/h.

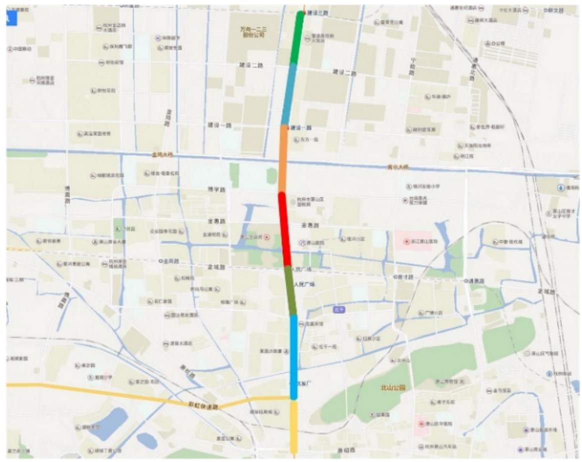


FIGURE 7. Prediction performance of TCA, DCNN, TCA-DCNN with varying interval lengths.

C. BENCHMARKS

The performance of TCA-DCNN is compared with the following models:

1) **LSTM (Long Short-Term memory) [22]** is an improvement of RNN by combining the interaction of the road network in both time domain and spatial domain.

2) **CNN (Convolutional Neural Network) [1]** has the benchmark framework similar to DCNN, but removes the deformable kernel. In addition, the input of the CNN benchmark is used as the spatio-temporal input matrix of DCNN.

D. THE EVALUATION METRICS

The experiment adopts two different metrics to evaluate the performance of the prediction model. They are Mean Absolute Error (MAE) and Mean Absolute Percentage

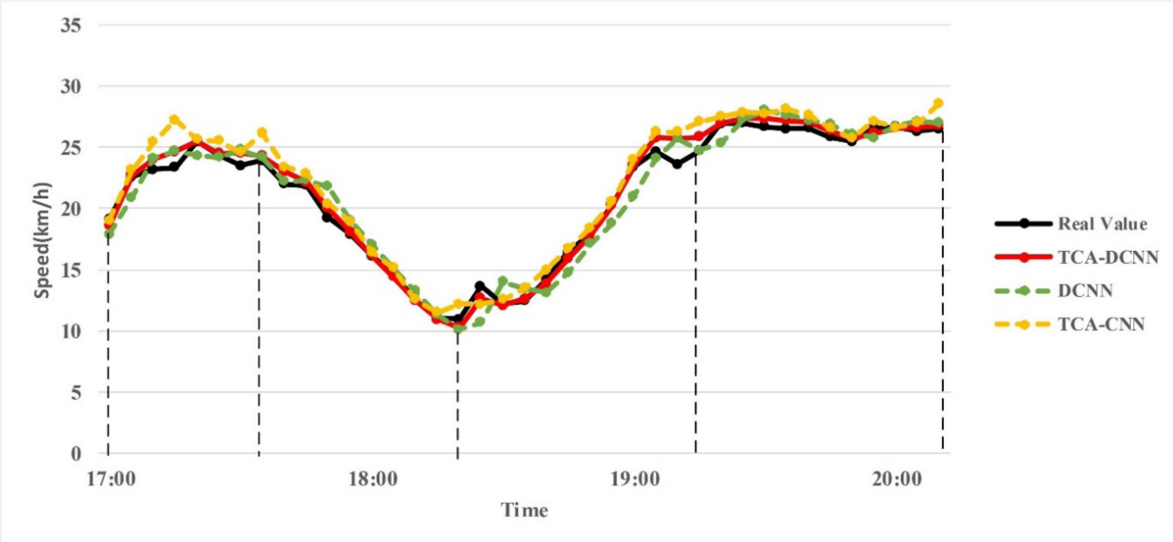


FIGURE 8. The real value and predicted value of traffic speed.

Error (MAPE) presented as following:

$$MAE = \frac{1}{N}\sum_{t=1}^{N} |observed_t - predicted_t|. \tag{9}$$

$$MAPE = \frac{1}{N}\sum_{t=1}^{N} \frac{|observed_t - predicted_t|}{observed_t}. \tag{10}$$

In the formula (9) and (10), $observed_t$ denotes real traffic speed value at time t, $predicted_t$ denotes predicted value at time t. N refers to the number of samples.

E. THE RESULT OF TIME PARTITION AND TEMPORAL CLUSTERING

As mentioned before, the entire time slot in this paper is from 0:00 to 24:00 on every day. The experiment uses partition time algorithm based on differential evolution to partition entire time slot into five sub-time slots as [0:00-6:00], [6:00-9:30], [9:30-12:00], [12:00-16:30], [16:30-21:45], [21:45-24:00]. For instance, [16:30-21:45] is chosen as prediction time for temporal clustering, and seven clusters are got. Among seven clusters the largest length of clusters has 41 days, the second largest one of clusters has 25 days, and the length of the remaining clusters are 15, 5, 1, 2 respectively.

F. OVERALL PERFORMANCE

We compare the performance of TC-DCNN with Benchmarks in interval length from 5 to 30 minutes. As shown in Fig.6. From the figure, we can find that the prediction error of TCA-DCNN is smaller than LSTM and CNN in two evaluation metrics. Besides, the figure demonstrates that the performance becomes worse when the interval length increases. What's more, it is a strange phenomenon in the fig.6(b) that MAPE of LSTM is dramatically rising from 5 interval lengths to 30 interval lengths. We think the reason is that the correlation decreases with the increase of the interval. LSTM cannot effectively extract correlation between the two consecutive periods of traffic speed decreasing with the increase of the interval length, especially when traffic speed decreases significantly.

In summary, compared with LSTM and CNN, TCA-DCNN has better performance, especially in long interval lengths prediction. The results of above experiment indicated that TCA can obtain similar traffic speed variation patterns in the same time slot of different days and DCNN can adaptively change convolution kernel to extract the spatio-temporal features from the traffic speed data effectively. Thus, the temporal clustering analysis and the deformable convolutional kernel are the key to success.

G. PERFORMANCE WITH TIME CLUSTERING ANALYSIS AND DEFORMABLE KERNEL

In this paper, the proposed model combines the temporal clustering with DCNN to predict the traffic speed. In order to evaluate the performance with temporal clustering and deformable kernel respectively, further simulation experiments are carried out on three prediction models, TCA-DCNN, DCNN and CNN. In this experiment, the prediction interval length is also set between 5 and 30 minutes.

As shown in the fig.7, the prediction errors of TCA-CNN and DCNN are similar at the same interval lengths from 5 to 35 mins. Besides, the prediction error of TCA-DCNN is the lowest in each prediction interval lengths. This indicates that there have been great improvements in the temporal clustering analysis and deformable convolutional, especially in long prediction period lengths.

To further demonstrate the performance of deformable kernel of DCNN, Fig.8 plots the real traffic speed and the predicted value of TCA-DCNN, TCA-CNN and DCNN. As can be seen from fig 8, from 17:00-20:00, the predictive interval length is 5 mins. The figure indicates that the combination of TCA and DCNN has better performance than only one of them.

VI. CONCLUSION AND FUTURE WORK

In this paper, a method called TCA-DCNN for traffic speed prediction is proposed. By combining differential evolution algorithm (DE) and hierarchical clustering (HC), TCA obtains several data clusters that contain consistent traffic speed variation pattern. By considering traffic environment, DCNN uses deformable kernels to adaptively adjust the size of convolutional kernel to extract different spatio-temporal information influenced by traffic environment to improve the prediction accuracy. The simulation experiment on real traffic data in Hangzhou, China proves that the model is effective.

The model proposed in this paper focus on traffic speed prediction without considering the challenges from traffic events like traffic accidents and human activities. Therefore, the prediction accuracy is greatly affected by traffic events. For the future work, we try to add feedback layer to perceive the prediction errors originated from the significant variation of traffic speed.

REFERENCES

[1] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A deep learning method," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, USA, Dec. 2017, pp. 499–508.

[2] D. Zang, J. Ling, J. Cheng, K. Tang, and X. Li, "Using convolutional neural network with asymmetrical kernels to predict speed of elevated highway," in *Proc. Int. Conf. Intell. Sci.*, Shanghai, China, 2017, pp. 212–221.

[3] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intell. Transport Syst.*, vol. 11, no. 9, pp. 531–536, 2017.

[4] Q. T. Tran, Z. Ma, H. Li, L. Hao, and Q. K. Trinh, "A multiplicative seasonal ARIMA/GARCH model in EVN traffic prediction," *Int. J. Commun. Netw. Syst. Sci.*, vol. 8, pp. 43–49, Apr. 2015.

[5] X. Kong *et al.*, "Mobility dataset generation for vehicular social networks based on floating car data," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 3874–3886, May 2018.

[6] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "LoTAD: Long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[7] X. Kong, F. Xia, J. Wang, A. Rahim, and S. K. Das, "Time-location-relationship combined service recommendation based on taxi trajectory data," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1202–1212, Jun. 2017.

[8] M. Ahmed and A. Cook, "Analysis of freeway traffic time-series data by using Box-Jenkins techniques," *Transp. Res. Rec*, vol. 773, no. 722, pp. 1–9, 1979.

[9] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, 1996.

[10] B. Williams, "Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1776, no. 1, pp. 194–200, 2001.

[11] H. Chen and S. Grant-Muller, "Use of sequential learning for short-term traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 9, no. 5, pp. 319–336, Sep. 2001.

[12] S. I.-J. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *J. Transp. Eng.*, vol. 129, no. 6, pp. 608–616, 2003.

[13] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[14] Y. Zou, X. Hua, Y. Zhang, and Y. Wang, "Hybrid short-term freeway speed prediction methods based on periodic analysis," *Can. J. Civil Eng.*, vol. 42, no. 8, pp. 570–582, 2015.

[15] J. Huang, Y. Bo, and H. Wang, "Electromechanical equipment state forecasting based on genetic algorithm—Support vector regression," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8399–8402, Jul. 2011.

[16] K.-Y. Chen, "Forecasting systems reliability based on support vector regression with genetic algorithms," *Rel. Eng. Syst. Saf.*, vol. 92, no. 4, pp. 423–432, Apr. 2007.

[17] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1817–1824, Nov. 2008.

[18] X. Xu, K. Zheng, D. Li, and Y. Yang, "New chaos-particle swarm optimization algorithm," *J Commun.*, vol. 33, no. 1, pp. 24–37, Jan. 2012.

[19] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[20] 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, p. e0119044, 2015.

[21] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

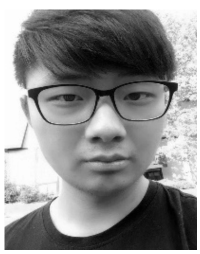
[22] Z. Zhao *et al.*, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, 2017.

[23] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2018, pp. 764–773.

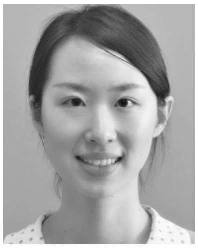
[24] S. Jeon and B. Hong, "Monte Carlo simulation-based traffic speed forecasting using historical big data," *Future Gener. Comput. Syst.*, vol. 65, pp. 182–195, Dec. 2016.

GUOJIANG SHEN received the B.Sc. degree in control theory and control engineering and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1999 and 2004, respectively. He is currently a Professor with the College of Computer Science and Technology, Zhejiang University of Technology. His current research interests include Artificial intelligence, theory, Big Data Analytics, and intelligent transportation system.

CHAOHUAN CHEN received the bachelor's degree in computer science technology from China Jiliang University, Hangzhou, China, in 2016, He is currently pursuing the master's degree in computer technology from the Zhejiang University of Technology. His current research interests include Big Data Analytics and intelligent transportation system.



QIHONG PAN received the bachelor's degree in software engineering from the School of Software Engineering, Beijing Jiaotong University, Beijing, China, in 2015. He is currently pursuing the master's degree in electrical and computer engineering with the Colorado State University, Fort Collins, CO, USA. His research interests include big data and machine learning.



SI SHEN received the B.S. degree in computer science and technology from Luoyang Normal University in 2011, and the M.S. degree from the Criminal Investigation Police University of China, Shenyang, China, in 2014. She is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Zhejiang University of Technology. Her current research interests include intelligent transportation and artificial intelligence.

ZHI LIU was born in Luoyang, China, in 1969. She received the B.S. degree in automatic control and the M.S. degree in system engineering from Xi'an Jiaotong University, Xi'an, China, in 1991 and 1994, respectively, and the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2001. She is currently a Professor with the College of Computer Science and Technology, Zhejiang University of Technology. She is a member of the China Computer Federation. Her current main research direction is intelligent transportation system.