An Efficient Traffic Prediction Model Using Deep Spatial-Temporal Network

Jie Xu[1,2(✉)], Yong Zhang[1,2], Yongzheng Jia[3], and Chunxiao Xing[1,2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China
xujl5@mails.tsinghua.edu.cn,
{zhangyong05,xingcx}@tsinghua.edu.cn
[2] Research Institute of Information Technology, Beijing National Research Center for Information Science and Technology, Beijing, China
[3] Institute of Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
jiayzl3@mails.tsinghua.edu.cn

Abstract. Recently years, traffic prediction has become an important and challenging problem in smart urban traffic computing, which can be used for government for road planning, detecting bottle-neck congestions roads, pollution emissions estimating and so on. However, former data mining algorithms mainly address the problem by using the traditional mathematical or statistical theories, and they were impossible to model the spatial and temporal relationship simultaneously. To address these issues, we propose an end-to-end neural network named C-LSTM to predict the traffic congestion at next time interval. More specifically, the C-LSTM is based on CNN and LSTM to collectively capture the spatial-temporal dependencies on the road network. Inspired by the procedure of handling the image by CNN, the city-wide traffic maps are first converted into a series of static images like the video frame and then are fed into a deep learning architecture, in which CNN extracts the spatial characteristics, and LSTM extracts the temporal characteristics. In addition, we also consider some external factors to further improve the prediction accuracy. Extensive experiments on reality Beijing transportation datasets demonstrate the superiority of our method.

Keywords: Road network · Traffic prediction · Residual · CNN · LSTM

1 Introduction

Road traffic prediction has become an interesting and challenging issue recently years, which is very important for the government in city managing, such as road planning, detecting the bottle-neck congestions roads, estimating the pollution emissions and so on. However, traditional traffic prediction studies were mainly based on statistical theory or mathematical theory, whose scalability and migration were poor, and they also tended to ignore the dynamic changes of each road segment, thus neglect the whole city-wide dependencies. Fortunately, with the great achievement of deep

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 386–399, 2019.
https://doi.org/10.1007/978-3-030-12981-1_27

learning in computer vision and natural language processing domains [1, 2], numerous researchers also implement the deep learning techniques to address the road transportation problems [3, 4, 5, 6, 7], since the deep learning can theoretically model complex nonlinear relationship. However, they have following disadvantages: In [7], they simply divided the historical data into categories groups and then fused them directly, lacking of theoretical proof for periods changes and an in-depth description for the temporal characteristics. In [4], they did not consider the impact of external factors (weather, Chinese festivals, etc.) on traffic flow, and the histories time sequences trajectories were too small.

In this paper, we predict the road traffic at next interval by using convolutional neural networks (CNN) and long short-term memory (LSTM) [8] to capture complex spatial and temporal nonlinear correlation, that means given a set of historical traffic data through a time period, the deep learning structure treats the traffic volume and flow as pixel values within a series of image, and predicts the traffic image like a motion-prediction issue. CNN and LSTM are applied to hierarchically learn the spatial-temporal relationship. There are three characteristics in the traffic trajectories data [6, 7]. (1) Spatial characteristic. A road traffic conditions will affect the relative link road traffic, for example, traffic accidents on overpasses may affect many connected roads traffic. At the other hand, the residential areas traffic perhaps affects the corresponding commercial areas traffic, the reverse is the same. (2) Temporal characteristic. Traffic conditions vary greatly with seasonal changes, or from rush hour to midnight, or from weekdays to weekend. (3) External factors, such as weather condition, time-of-day, day-of-week and Chinese festivals which are proven to be promising impact factors. In summary, the contributions of this work are summarized as follows:

First, in this paper, we illustrate how a road network traffic condition can be transferred to a image-related heat map (in particular, CNN and LSTM related), which is helpful for deep learning methods to describe the road traffic features.

Second, we propose an end-to-end combination model named C-LSTM which is comprised of CNN and LSTM to sketch the road network traffic image characteristic. CNN utilize the strengthen to capture the spatial characteristic, while LSTM capture the temporal characteristic of traffic map. In order to make the model fully close to reality, we fuse the external factor such as weather and time metal date, which can obvious improve forecasting effectiveness.

Third, we conduct extensive experiments on real datasets, and results show the high convergence rate and advancement of the our approach.

The rest of this paper is organized as follows: We propose the related work in Sect. 2, then model the problem and illustrate our algorithm in Sect. 3. Experimental results are discussed in Sect. 4. The conclusions are summarized in Sect. 5.

2 Related Work

There are some previously existing works on predicting time series movements [9] based on the history trajectories, the approaches can be classified into mainly two categories: traditional algorithms and deep learning algorithms.

2.1 Traditional Prediction Algorithms

Shu et al. [2] proposed an adapted traffic prediction methods named FARMIMA, the dependencies in the network traffic can be divided into two kinds, i.e., long-range and short range dependence. They adjusted an indicator parameter named weight bias for short and long range dependence by congregating on standard autoregressive integrated moving average. Clark et al. [10] designed an intuitive method for forecasting the traffic by using a pattern matching technique that exploited the three-dimensional nature of the traffic state. The approach was a multivariate extension of nonparametric regression and can be easily understood by the practitioners. However, they neither took consideration of the main factors of traffic conditions, nor considered the interaction among different city-wide regions. Min et al. [11] proposed a framework which took into account the spatial characteristics of a road network, that can reflect not only the distance but also the average speed on the road segment, they also involved incorporating weather, incident data, current or planned roadwork into the forecasting model. However, all above algorithms fail to depict the spatial and temporal relationship simultaneously.

2.2 Deep Learning Algorithms

Recently, a series of studies applied CNN and LSTM to capture spatial and temporal dependencies and achieved great success [3, 4, 5, 6, 12]. Ke et al. [6] proposed a one end to end convolutional and multiple LSTM network named FCL-NET to address these three dependencies for short-term passenger demand forecasting. The evidence from benchmark models proved that spatial-temporal correlations in models can greatly improve the predictive accuracy. Zhang et al. [7] introduced a deep learning methods based on CNN named ST-ResNet for citywide crowd flows prediction. They divided the historical data into three categories, i.e., closeness, period, trend data, which depicted the denoting recent time, near history and distant history respectively. The residual unit were applied for training super deep neural networks. Yu et al. [4] designed a novel deep architecture with CNN and LSTM, namely, spatiotemporal recurrent convolutional networks (SRCNs) for traffic prediction. They divided the road network into many grids whose average velocity represented the each grid velocity in specified timestamp, that can retain the fine-scale structure of a transportation network, but they didn't consider the extensive features. Wang et al. [3] proposed an end-to-end deep learning framework named DeepTTE to estimate the travel time of the whole path directly, in which the geographic information were integrated into the classical geo-convolution, which was capable of capturing spatial correlation. Their method was novel and worked well for depicting vehicle trajectories. Zhang et al. [14] presented a deep spatial-temporal neural networks named FCN-rLSTM to sequentially count vehicles from low quality videos on road network, that model connected fully convolutional neural networks (FCN) with LSTM in a residual learning fashion, and enabled the processing to refine the feature representation and implement a novel end-to-end trainable mapping from pixels to vehicle count.

3 Traffic Prediction Based on Deep Learning

3.1 Preliminaries

Definition 1 Road network G = (V, E), which is used to model a road network, where a vertex set V is associated with a geographical position, including the longitude and latitude, a vertex set E is defined for the edges between two positions. The road network is divided into multiple same size grids (e.g. 10 * 10 m), where the velocity in each region is regarded as the same.

Definition 2: Average velocity $\mathbf{V}_{ij}^{t}$ is defined as the average ratio of travel distance to travel time in i * j region according to the road links condition at specified timestamp t, in our paper, the length of the time interval is set as 30 min.

Definition 3: Velocity heat map. Give a set of historical trajectories, we can treat the velocity as pixel values in one grid, and mark different colors according to different velocities. By doing so, the whole velocity heat map can be built based on the road network and can be denoted as V(R, t) at time t. As in Fig. 1, the left figure represents the whole traffic status, the different colors lines on the road represent different velocities, the right figure represents how the velocity of each region is mapped and colored in corresponding region.

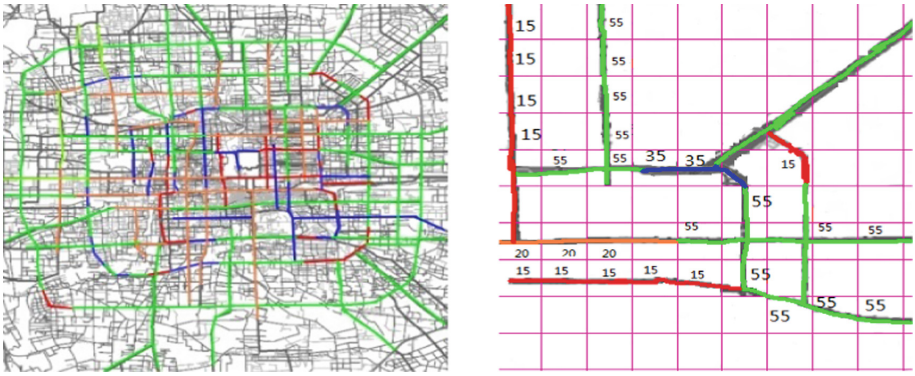

Fig. 1. Velocity heat map

Problem: Give a time series of historical road trajectories maps V(R, t − 3), V(R, t − 2), V(R, t − 1), our goal is to predict the next V(R, t).

3.2 Framework of C-LSTM

Figure 2 presents the architecture of C-LSTM, which is comprised of three major components [13]. For geographic problems, the key spirit is how to transform the problem into a sequence of 2D image, i.e., velocity heat map, which is similar with the video frame, the image itself has natural spatial and temporal attributes which is

associated with CNN and LSTM. The strength of CNN is to outline spatial characteristic, on the other hand, for traffic information, intuitively, we can find that traffic conditions vary with rush hours and midnight, weekdays and weekend, using RNN (recurrent neural network) to describe temporal series characteristics has achieved great success. In this paper, we utilize the CNN and LSTM to jointly predict traffic in a residual fashion, such combination leverages the strengths of Resnet (Deep residual network) [15] for pixel-level description and the strengths of LSTM for learning temporal dependencies, we further aggregate the external components to describe the influence of complex external factors.

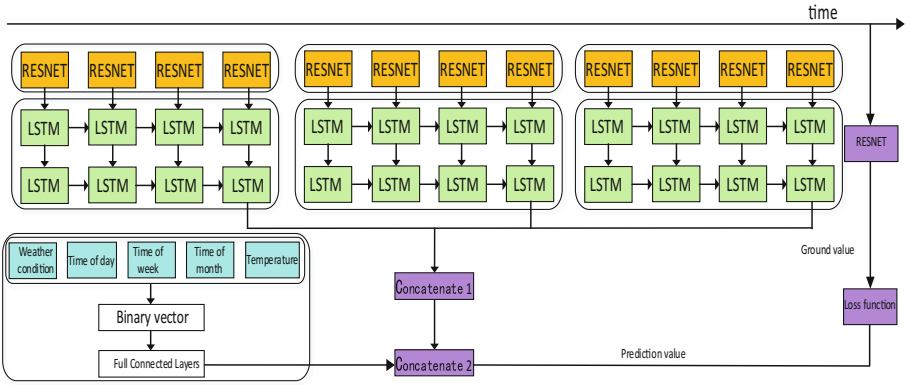

Fig. 2. Framework of C-LSTM

Steps for Model: The trajectories data are divided into three groups according to time axis, denoting as recent time, week period, and month period respectively. In theory, shallow CNN can capture near-distance features, and deep CNN can capture far away area information, thus each image characteristics can be exacted by a convolutional neural network with the residual units [4, 16]. To predict the heat map V(R, t), then the LSTM capture the temporal sequences, the output of each group of CNN is used as the input of LSTM. Theoretically, more stacked LSTM layers can overcome the vanishing gradient and exploding gradient problems [17], in our model, the LSTM model was stacked by two to four layers.

For external component, the external factors are collected and organized at the same time interval, such as weather, time of day, time of week, etc. External factors are first transfer to binary vector and feed into full connected layers, whose output are fused with the output of LSTM. Finally, the aggregation is mapped into one-dimensional vector whose value is restrict within $[-1, 1]$, by steadily narrowing the loss function to convergence status.

3.3 Spatial Characteristics Captured by CNN

The convolution neural network (CNN) has achieved great success in extracting features. In the road network structure, a snapshot of network traffic flow at specified time

can be seen as a 2D image, the traffic conditions on a road not only affect the most adjacent area, but also have potential impact on more distant regions. CNN is good at processing image and video frame with both near distance and far away distance correlations, the input for CNN is a tensor for sequence 2D heat map at a pixel level, the value is converted from −1 to 1 with normalization technique.

In ours study, the deep convolution neural networks is firstly constructed to capture the spatial relationships in the heat road traffic image. As in Fig. 2, we use zero padding to the end of the boundaries of city. The regular transformation at each CNN layer is defined as follows:

$$O_k^l = f(w_k^{l-1} * O_k^{l-1} + b_k^{l-1}) \tag{1}$$

Where the $f(\cdot)$ denotes an activation function and * denotes the convolutional operation, which usually is a nonlinear activation function. The w_k^l presents weight parameters matrix, the O_k^l represents the output of the k filter for l-th layer, in our study, the activation function is tanh (x).

From [7], near spatial dependencies can be seized by the shallow CNN, however, since one convolutional layer can only figure out near characteristic because of the limitation of kernel size, we need design deep CNN layers, e.g. 50 layers, to capture the distance dependencies. However, it is also well-known that very deep convolutional networks exposed a degraded problem, the accuracy becomes saturated and inaccurate, and more deep layers may generated higher error.

In this paper, the residual method is employed to train the CNN. The Resnet take advantage of a connection method named "shortcut connection", and bottleneck design, whose purpose is to reduce the number of calculations and parameters. As in Fig. 3, shortcut connections are inserted with the plain network [15] by residual blocks. The right figure is called "bottleneck building design". To reduce the number of parameters, the first 1 × 1 convolution reduces the 256-dimensional to 64-dimension, and then recovers at the end through a 1 × 1 convolution with 64 filters. For plain

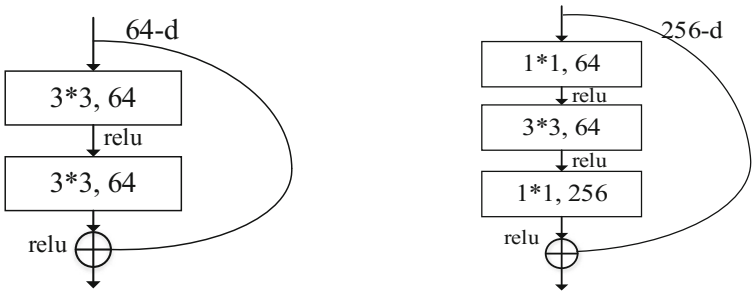

Fig. 3. Residual unit

network, it can be applied in networks of 34 layers or less as in left Fig. 3, and shortcut for bottleneck design is usually implemented in deeper networks like 50/101/152.

The updating rule of our residual layer can be expressed as:

$$X_k^{l+1} = X_k^l + F(X_k^l, \theta_k^l) \tag{2}$$

Where $F(\cdot)$ is the residual function as presented in Fig. 3. After convolution, max-pooling is applied to select remarkable features, and fully connected layer is to generate the class scores for different features at each interval t, the output features vector $X \in R^{1000}$ is fed as the LSTM input.

3.4 Temporal Characteristics Captured by LSTM

Intuitively, for the road network velocity, such as the rush hour and midnight of the day, the heat map shows a periodicity changes, similar phenomena occur in the time of week and time of month. For example, 5 pm, on Monday in June, the traffic condition is similar to 5 pm, on Monday in July.

The one of most successful model for handling time sequential is RNN (recurrent neural network), which is a repeated structure and the output of a neuron can be used as the input for the next neuron unit. The main components can be concluded as $s_t = f(Ws_{t-1} + Ux_t)$, where the f(•) function is usually a nonlinear function and x_t is the input at time t. However, RNN can only process a certain length of sequence, if the time interval is too large, there may be gradient disappearance or exploding gradient problem, and the effectiveness becomes poor. Consequently, the LSTM based on processing the long-term information method is introduced [18], which is a special structural variant of RNN. The LSTM designs one forget gate, one input gate, and one output gate. The gates record and pass the information through the units. In this way, this gate can memorize the information that needs to be saved, or drop redundant and irrelevant one. The structure parameters for all gates are same and shared through whole steps.

At each time interval, LSTM takes time series C_t as an input, and then all information is accumulated to memory cell, the architecture of LSTM is defined as follows:

$$f_t = \delta(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \tag{3}$$

$$i_t = \delta(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \tag{4}$$

$$\widetilde{C}_t = tanh(W_{Ch}h_{t-1} + W_{Cx}x_t + b_C) \tag{5}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \tag{6}$$

$$o_t = \delta(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \tag{7}$$

$$h_t = o_t \circ tanh(C_t) \tag{8}$$

Where the W_f, W_i, W_c, W_o are the weight parameters matrices and the b_f, b_i, b_c, b_o are the biases values of the three gates, the δ denotes the non-linear activation function. In our case, the LSTM with CNN are combined to jointly learn traffic velocity heat map.

Furthermore, it has been shown that multiple stacked LSTM layers are more efficient to increase the model capacity compared with a single LSTM layer. In our model, the input sequence of the LSTM is the features (i.e. 1000 dimension) outputted by the last layer of Resnet. For different temporal period, the number of time series is set 3, 3, 4 respectively. The three different CNN-LSTM units are integrated by using the fusion methods as follows:

$$X_t^{Lstm} = W_{h1} \circ h_t^h + W_{h2} \circ h_t^d + W_{h3} \circ h_t^m \tag{9}$$

Where the $h_{t+1}^h, h_{t+1}^d, h_{t+1}^m$ represent the hour dependencies, day dependencies, week dependencies respectively, $\bigcirc$ is Hadamard product.

3.5 The Structure of External Factors

Traffic flows can be affected by many complex external factors, such as weather and event. For instance, rainy days are usually more congested than usual, and road is more prone to have high level crowd [19], etc. In our implementation, we mainly consider weather condition, day-of-hour, time-of-week, and day-of-month. Note that these property values cannot be directly fed into CNN, we embed the weather condition as $X \in \mathrm{R}^{16}$, time-of-hour as $X \in \mathrm{R}^{24}$, day-of-week as $X \in \mathrm{R}^{7}$, and day-of-month as $X \in \mathrm{R}^{12}$ by using the hot coding to transform each categorical attribute into a vector, then concatenate the individual vectors into a integrated vector, further feed them into a full FC connection layer to reduce the dimension of spatial-temporal features [20]. The output of LSTM is fused with the external component. The definition is formulated as follows:

$$\widetilde{\mathrm{Y}}_t = \sigma(W_{Lstm} \circ X_t^{Lstm} + W_{ext} \circ E_{ext}) \tag{10}$$

$\sigma(\cdot)$ is a sigmoid function defined as $\sigma(x) = 1 = (1 + e^{-x})$, W_{Lstm} and W_{ext} are two learnable parameter sets. For the actual value of traffic heat map at time interval t, we also capture the feature using the same Resnet architecture, and then fed them into FC layers, whose output is $\mathrm{Y_t}$, and the dimensions is equal to the X_t^{Lstm}. In this paper, recall that our goal is to predict road congestion at the next time interval t, we need to reduce the error between the actual value and the predicted value within a reasonable deviation during training process, thus the loss function is defined as follows:

$$L(\theta) = \left\| Y_t - \widetilde{\mathrm{Y}}_t \right\| \tag{11}$$

Where θ are all learnable parameters needed to be trained. We continuously adjust the parameter sets by Tensorflow until loss function converges.

4 Experiments and Discussions

4.1 Dataset

Datasets: We use two real historical data set [21] in Beijing road which contains about 330,000 vertices and 440,000 edges, the **Taxi data**, which contains about 180,000 trajectories generated by more than 7,000 public taxicabs, the **Ucar data**, which contains about 480,000 trajectories generated by more than 5,000 public taxicabs. Those abnormal records are first filtered out, and the map matching algorithm is employed to locate the deviated GPS data to the road network, the vehicle velocity maps are converted to a congestion heat map at every 15 min interval.

Meteorological data: We record the Beijing weather data from Beijing Meteorological Bureau to incorporate the impacts on the road traffic. The weather conditions are divided into 16 types: normal days, rainy, sunny, snowy, overcast, cloudy, sleety, Foggy etc. [22]. For example, we choose the following nine features, e.g. 2016-06-09, the Chinese Dragon Boat Festival, 10 am, hourly temperature, rain, wind speed, as one hot-encoded vector that denotes external factors.

4.2 Parameters Setting

Parameters setting: The parameters are described as follows. In the Resnet component, first, the global road network is divided into small equal region with size 10 * 10 m, the layers of Resnet is set as 32/50/101, with core kernel size 3 * 3, the dimensional of time series C_t for LSTM is set as 1000. Our model is implemented with Python 2.0. We adopt Adamax optimization algorithm to train the parameters, the learning rate is 0.1, the weight of loss is 0.01. The embedding function converts the raw data to the range of [0, 1] by using max-mix normalization, the formula is defined as follows:

$$Xnorm = \frac{X - Xmin}{Xmax - Xmin} \tag{12}$$

4.3 Baseline Algorithms

To demonstrate the validity of our model, we compare it with 6 baseline methods, including:

ARIMA [2]: ARIMA means AutoRegressive Integrated Moving Average, which is a class of statistical models and can captures a suite of different standard temporal structures in time series data, leverage the dependency between an observation and some number of lagged observations with a residual error.

GBDT: GBDT is a machine learning technique for regression, classification and sorting tasks by ensemble multiple weak learners, usually decision trees. It belongs to ensemble learning.

SRCNS: SRCNS is a spatiotemporal recurrent convolutional networks, which inherit the advantages of deep convolutional neural networks (DCNNs) and the long

short-term memory (LSTM) neural networks, the DCNN captures the spatial dependencies, and LSTM captures the temporal dependencies.

DMVST-Net: DMVST-Net is a deep multi-view spatiotemporal network. To model spatiotemporal relationships [5], they construct a region map based on the similarity of demand patterns for modeling related but distant areas.

FCL-Net [6]: FCL-Net is a fusion convolutional long short-term memory network to forecast an on-demand ride service, this model is stacked and fused by convolutional operators, LSTM layers, multiple conv-LSTM layers. A tailored spatially random forest is utilized to score the variables for feature selection.

ST-ResNet [7]: In ST-ResNet, the historical data was divided into three categories, they leveraged the residual neural network framework to model the time tightness, period and trend characteristics of crowd traffic respectively, and also added external attribute information such as weather, time of day, time of month, time of week. In this paper, the above algorithms are all implemented under the same equipment and environment.

4.4 Evaluation Metric

In our study, the Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE) are employed as evaluation metric, the definitions are as follows:

$$MAPE = \sum_{t=1}^{n} \left| \frac{y_t - \tilde{y}_t}{\mathrm{y}_t} \right| * \frac{1}{n} \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} (y_t - \tilde{y}_t)^2} \tag{14}$$

Table 1. Performance comparisons

Model	Taxi		Ucar	
	MAPE	RMSE	MAPE	RMSE
ARIMA	0.32	51	0.28	47.25
GBDT	0.278	43	0.257	39.41
SRCNS	0.243	39	0.214	35.47
DMVST-Net	0.196	28.74	0.176	24.19
FCL-Net	0.174	24.68	0.154	26.56
ST-ResNet	0.157	18.4	0.137	13.61
Ours	**0.123**	**13.34**	**0.104**	**8.76**

Where y_t is the predictive value, the $\tilde{\mathrm{y}}_t$ is the actual value. To verify the effectiveness of C-LSTM, we compare it with several state-of-the-art methods, the results are shown in Table 1.

From the Table 1, we can see that the MAPE and RMSE of the ARIMA perform poor results(i.e., has a MAPE of 0.32 and RMSE of 51, respectively), which means that the simple traditional prediction method cannot effectively describe the spatial-temporal information. The effect of DMVST-Net and FCL-NET is similar, and achieves better performance than ARIMA and GBDT, the comparison shows that deep learning methods can work. The ST-ResNet shows a good performance, however, it does not illustrate the time dependencies very well. Our algorithm significantly outperforms above mentioned methods with the lowest MAPE (0.123% and 0.104%) and RMSE (13.34 and 8.76) on two datasets, it verifies the superiority and feasibility of the our approach, as our algorithm further exploits LSTM and takes account of the influence of external factors.

4.5 Effectiveness of Resnet

In this section, we compare the model performances with different Resnet varieties, the traditional AlexNet [23] network is used to replace the Resnet. We can observe that the result of the replacement is not as well as Resnet. On the other hand, we compare different Resnet variants with respect to different number of layers [15], i.e., 34/50/101.

Table 2. Experimental results with different CNN variants

Model	Taxi		Ucar	
	MAPE	RMSE	MAPE	RMSE
AlexNet	0.235	17.63	0.215	14.86
Resnet 34	0.176	15.89	0.164	11.34
Resmet 50	0.153	15.14	0.132	9.72
Resmet 101	**0.123**	**13.34**	**0.104**	**8.76**

The results are shown in Table 2, we can see that as the layers increases, the values of MAPE and RMSE decay, the Resnet 101 decreases 36% and 22% for MAPE and RMSE respectively comparing with Resnet 34 for Ucar dataset.

4.6 Effectiveness of LSTM

In our article, for different sequence length for LSTM, we mainly consider the number of days in one week and the number of hours in one day. In fact, Multiple layers of LSTM are added, such as Layers 3 and 4, to evaluate the effect. The results reveal that the effect of the 3 and 4 layers are better than that 2 layers, but the increasing trend is not obvious. In addition, we set $(d_1, d_2, d_3) = (3, 5, 7)$, and $(h_1, h_2, h_3, h_4) = (2, 4, 8, 16)$ which indicates that traffic speeds are predicted from the previous (3, 5, 7) days and (2, 4, 8,16) hours based on historical data by fixing the number of other two LSTM hidden units components respectably. As in Table 3, we observe that the layers-h-8 yields the lowest MAPE and RMSE, which demonstrates when the historical data length goes larger, the prediction error decreases. However, for layers-h-16, it no

Table 3. Experimental results with different hidden units of LSTM.

Model	Taxi		Ucar	
	MAPE	RMSE	MAPE	RMSE
Layers-d-3	0.139	16.78	0.121	11.34
Layers-d-5	0.135	15.67	0.116	10.67
Layers-d-7	0.127	14.32	0.112	9.87
Layers-h-2	0.153	16.89	0.137	13.24
Layers-h-4	0.147	14.78	0.119	10.14
Layers-h-8	**0.123**	**13.34**	**0.104**	**8.76**
Layers-h-16	0.148	15.64	0.121	10.24

longer displays remarkable results, we find that it tends to overfit the training data and thus exhibits the slightly degrades prediction performance.

4.7 Effectiveness of Attribute Component

In our article, we mainly leverage external influence conditions, but in fact, each external condition is not always available at the same time. The whole external information component is removed first, the rustles decrease nearly 30%. Second, we choose frequently-used available external conditions as attribute information, excluding partly uncommon factors such as temperature, wind speed, humidity, etc., the results display that the prediction error increases, but the degree is not large, this finding confirms that the weight of external factors are different.

5 Conclusions

In this paper, we propose a novel deep learning end-to-end model method based on CNN and LSTM model for predicting the future traffic flow based on real historical traffic data. The method inherits the strength of both CNN and LSTM, and transform historical data into heat map firstly, then employ CNN to extract the spatial features, further utilize LSTM to capture the temporal characteristic. To validate the effectiveness of the proposed C-LSTM, six previous prediction approaches are exploited to compare the results by extensive experiments in terms of RMSE and MAPE, the results show that our method can effectively deal with spatiotemporal and spatial information, and demonstrate the superiority of our methodologies.

The key spirit of this paper is how to transform the historical trajectories data on road network into a heat map, and then take advantage of the deep learning method as the domain of video frame research, which can be used for these similar types transportation problem, such as taxi demand, traffic flow, POI (Point Of Interest) prediction and so on. For future work, we plan to (1) add some other mechanisms, such as attention mechanism, to improve the effectiveness in time sequence learning task, (2) consider incorporating the road semantic information for deep learning model,

(3) apply machine learning to interdisciplinary areas such smart transportation and economics disciplines.

Acknowledgments. This research was financially supported by NSFC (91646202), the National High-tech R&D Program of China (SS2015AA020102), Research/Project 2017YB142 supported by Ministry of Education of The People's Republic of China, the 1000-Talent program, Tsinghua University Initiative Scientific Research Program.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105. Hinton (2012)
2. Shu, Y., Jin, Z., Zhang, L., et al.: Traffic prediction using FARIMA models. In: IEEE International Conference on Communications, ICC 1999, vol. 2, pp. 891–895. IEEE (1999)
3. Wang, D., Zhang, J., Cao, W., et al.: When will you arrive? Estimating travel time based on deep neural networks. In: AAAI (2018)
4. Yu, H., Wu, Z., Wang, S., et al.: Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors **17**(7), 1501 (2017)
5. Yao, H., Wu, F., Ke, J., et al.: Deep multi-view spatial-temporal network for taxi demand prediction. arXiv preprint arXiv:1802.08714 (2018)
6. Ke, J., Zheng, H., Yang, H., et al.: Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. Transp. Res. Part C Emerg. Technol. **85**, 591–608 (2017)
7. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: AAAI, pp. 1655–1661 (2017)
8. Xingjian, S.H.I., Chen, Z., Wang, H., et al.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810. MLA (2015)
9. Zheng, Y.: Trajectory data mining: an overview. ACM Trans. Intell. Syst. Technol. (TIST) **6** (3), 29 (2015)
10. Clark, S.: Traffic prediction using multivariate nonparametric regression. J. Transp. Eng. **129** (2), 161–168 (2003)
11. Min, W., Wynter, L.: Real-time road traffic prediction with spatio-temporal correlations. Transp. Res. Part C Emerg. Technol. **19**(4), 606–616 (2011)
12. Song, X., Kanasugi, H., Shibasaki, R.: DeepTransport: prediction and simulation of human mobility and transportation mode at a citywide level. In: IJCAI, vol. 16, pp. 2618–2624 (2016)
13. Liao, S., Zhou, L., Di, X., et al.: Large-scale short-term urban taxi demand forecasting using deep learning. In: Proceedings of the 23rd Asia and South Pacific Design Automation Conference, pp. 428–433. IEEE Press (2018)
14. Zhang, S., Wu, G., Costeira, J.P., et al.: FCN-rLSTM: Deep spatio-temporal neural networks for vehicle counting in city cameras. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 3687–3696. IEEE (2017)
15. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

16. Ma, X., Tao, Z., Wang, Y., et al.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C Emerg. Technol. **54**, 187–197 (2015)
17. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
18. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM (1999)
19. Wang, J., Gu, Q., Wu, J., et al.: Traffic speed prediction and congestion source exploration: a deep learning method. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 499–508. IEEE (2016)
20. Zheng, Y.: Methodologies for cross-domain data fusion: an overview. IEEE Trans. Big Data **1**(1), 16–34 (2015)
21. Ta, N., Li, G., Zhao, T., et al.: An efficient ride-sharing framework for maximizing shared route. IEEE Trans. Knowl. Data Eng. (TKDE) **30**, 219–233 (2017)
22. Tong, Y., Chen, Y., Zhou, Z., et al.: The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1653–1662. ACM (2017)
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)