Building Effective Large-Scale Traffic State Prediction System: Traffic4cast Challenge Solution

Yang Liu[1,a,*], Fanyou Wu[2,b,*], Baosheng Yu[3], Zhiyuan Liu[1], and Jieping Ye[4]

[1]Southeast Univeristy, School of Transportation, Nanjing, China
[2]Purdue University, Department of Forestry and Natural Resource, West Lafayette, USA
[3]The University of Sydney, UBTECH Sydney AI Centre, School of Computer Science, Sydney, Australia
[4]DiDi Chuxing, Didi AI Labs, Beijing, China
[a]Email: 230179629@seu.edu.cn
[b]Email: wu1297@purdue.edu
[*]Equal Contribution

ABSTRACT

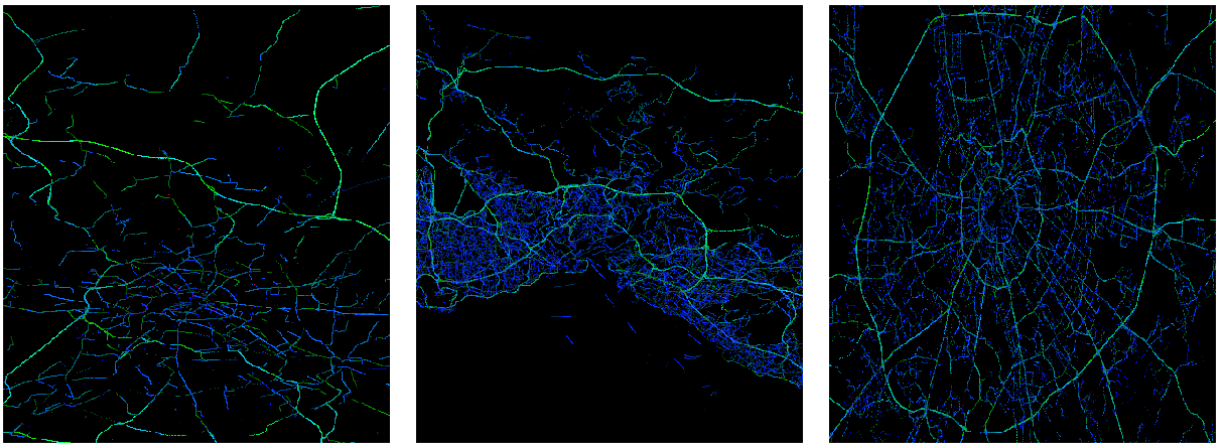
How to build an effective large-scale traffic state prediction system is a challenging but highly valuable problem. This study focuses on the construction of an effective solution designed for spatio-temporal data to predict large- scale traffic state. Considering the large data size in Traffic4cast Challenge and our limited computational resources, we emphasize model design to achieve a relatively high prediction performance within acceptable running time. We adopt a structure similar to U-net and use a mask instead of spatial attention to address the data sparsity. Then, combined with the experience of time series prediction problem, we design a number of features, which are input into the model as different channels. Region cropping is used to decrease the difference between the size of the receptive field and the study area, and the models can be specially optimized for each sub-region. The fusion of interdisciplinary knowledge and experience is an emerging demand in classical traffic research. Several interdisciplinary studies we have been studying are also discussed in the Complementary Challenges. The source codes are available in https://github.com/wufanyou/traffic4cast-TLab.

1 Introduction

Real-time traffic state prediction is an essential component for traffic control and management in an urban road network. The ability to predict future traffic state (e.g., flow, speed) can help improve traffic conditions, fleet organization, utilization rate, and social welfare[1,2]. Essentially, the traffic prediction is a time series problem, which is performed based on the changes of historical demand. A representative time-series prediction tool is the recurrent neural network (RNN), along with its diverse variants[3,4]. Apart from the temporal dimension, the correlation in the spatial dimension is also extensively incorporated by many works. Regions that are close to each other or share similar land-use structures may exhibit a homogeneous demand pattern. Techniques widely applied in computer vision like convolutional neural network (CNN)[2,5] and the emerging graph-based networks[6–9] are often adopted. Furthermore, multi-source data are also introduced in some literature to allow for the external influencing factors, such as weather conditions and neighboring points-of-interest[10,11].

2 Data Description and Problem Definition

The organizer provides industrial-scale real-world data for 3 full cities (Berlin, Istanbul and Moscow) throughout a year. The organizer divides the whole city into a 436×495 grid, each pixel of which represents a region of $100m \times 100m$. There are a totally of 288 frames each day, and each frame of the raw data represents 5-minutes aggregated information with three channels, *i.e.*, traffic volume, speed, and direction, respectively. The sample images are illustrated in Figure 1. The data of volume and speed are scaled to $[0, 255]$ through a min-max scaler.



Berlin Istanbul Moscow

Figure 1. Sample Images from traffic4cast data set for May 1st, 2018. In general, the greener the color, the heavier the road.

Regarding the direction data, they are obtained by binning probes into four heading directions of North-East $(0,90)$, South-East $(90,180)$, South-West $(180,270)$ and North-West $(270,359]$. In this way, the raw heading values $[0,359]$ are mapped as four distinct values of $\{1,85,170,255\}$ in the order of (NW,NE,SW,SE), and the major heading bin, *e.g.* the bin containing the largest number of points, is selected. Here, missing values are also represented by 0.

Problem: This challenge can be split into three sub-tasks, i.e., use the given data to predict speed, flow, and direction at each pixel separately in the given time intervals, which is a typical time series prediction problem. Pixel-wise mean squared error (MSE) is used to evaluate the performance for ranking the submitted prediction results.

3 Solution

3.1 Network

Increasing the receptive field. One important attribute of the data is the spatial correlation. This means that the traffic state in a specific region can be affected by the adjacent regions. For example, when traffic congestion occurs in a link of the urban road network, it might propagate upstream, thus causing traffic congestion in neighboring areas. To capture this phenomenon, a large receptive field is theoretically better. A simplest way to increase receptive field is to increase kernel size of convolutional operation. However, due to the limit of computing devices, increasing the receptive field through enlarging the convolutional kernels is not applicable. Therefore, pooling operation is a more feasible way to increase the receptive field. Considering that this task is similar to pixel-wise image segmentation, we adopt a structure similar to U-net[12]. Moreover, the inputs are cropped to further decrease the difference between the size of the receptive field and the study area (see section 3.3 for more details).

Use mask instead of spatial attention. To address the data sparsity, the introduction of spatial attention might be possible, but it will lead to a sharp increase in training time. Therefore, a *mask* is designed to improve efficiency. We concatenate one more channel to the last layer before the 1×1 CONV, which checks whether the mean of input is equal to zero or not.

3.2 Feature Engineering

Convert features into channels. Essentially, this task is a time-series prediction problem. For one-dimensional time-series prediction, a typical solution is to transform the problem into a supervised learning problem with the help

of feature engineering. Referring to this idea, the features are input into the model as different channels. Limited by the computing devices (*i.e.*, $4 \times$ 2080Ti GPUs), there are 101 input features (*i.e.*, 101 channels). More features can usually bring higher accuracy.

Time smoothness features. Time series data are intrinsically continuous rather than discrete. Hence, time series seldom produce sharp variations, which means that the traffic state at a certain time interval has a close resemblance to traffic state at the adjacent time interval. The feature crosses are also considered. Therefore, in each prediction subtask, the data of all three subtasks from the previous 12 time intervals are used. A total of 12×3 channels are used from $[T-12, T-1]$ with the same day, where T denotes the predicted time index, and the number 3 indicates the data of three subtasks.

Periodic features. The similarity between the traffic states of two different days can be attributed to the periodic characteristics of traffic states, which typically repeat every 24 hours. We use in total 16×3 channels from $\{D-14\} \bigcup [D-7, D-1] \bigcup [D+1, D+7] \bigcup \{D+14\}$ with the same time index, where D is the predicted day. To remove random noises, moving windows with the windows size of 13 are applied to smooth each time index.

Statistical features. We use in total 5×3 channels from 5 average given period on the same day.

Time-dependent features. Two channels in total are used to represent month and week average in the predicted time index and corresponding task.

3.3 Region Cropping

We further divide the study area into five sub-regions with size 299×299 (see Figure 2). Region cropping allows our model to be trained on weak computing devices. As mentioned above, it decreases the difference between the size of the receptive field and the study area. In addition, the models can be specially optimized for each sub-region, as the optimal parameters for the whole city may not be the best parameters suited for each sub-region. We spliced the prediction results of each sub-region into a complete region, and the overlapping portions were replaced with average values. To sum up, in our final solution, we trained in total $3^{\text{cities}} \times 3^{\text{subtasks}} \times 5^{\text{subregions}} = 45$ models. Here, it should be pointed out that our models can be fine-tuned and actually efficient in training.

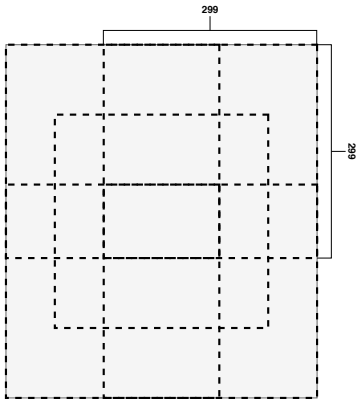


Figure 2. Schematic diagram of region cropping

3.4 Training

We train all models in following procedure. First, for each city, we train a model from scratch for direction prediction task of a 299×299 subregion. Then using those parameters as pre-trained models, we fine-tune all the rest models. The fine-tune step starts from a larger learning rate, more data, and no data augmentation to lower learning rate, fewer data, and more data augmentation. The training process is summarized in Algorithm 1 as below.

Algorithm 1 Training

for CITY **in** {Berlin, Istanbul, Moscow} **do**
  Train a model from scratch for direction prediction task with SUB_REGION= 0.
  for SUB_REGION **in** $[0,4]$ **do**
    Use the pre-trained model above, fine-tune the models for the other tasks.
  end for
  for SUB_REGION **in** $[0,4]$ **do**
    Fine-tune all the models with lower learning rate, fewer data, and data augmentation.
  end for
end for
return $\boldsymbol{x}$

4 Complementary Challenges

4.1 Alternative Metrics

Pixel-wise mean squared error (MSE) is used to evaluate the performance for the ranking of submitted prediction results. The MSE is not the idea loss for this data set due to the sparsity of data and the improper coding of the heading channel. For Moscow, the MSE of the heading channel is approximate 3.1×10^{-2} while speeding, and volume channel are 1×10^{-4} and 5×10^{-3} respectively. This magnitude difference also happened in Berlin and Istanbul. It is clear that the heading channel dominates overall scores comparing to the leader-board score difference in 10^{-4}. In other words, a model which predicts better heading channel will rank higher in the leader board. If we direct train the model with MSE loss, though the performance is the best regarding lead board ranking, the predicted values are meaningless. Table 1 is an example: our best model tries to predict large value more conservative (trend to out-put small values and made frame unrealistic and blurry) because of the punishment of square error. This characteristic hinders this model to real-world applications when detailed general directions (NW, NE, SW, SE) are more critical. In real-world traffic prediction systems, MSE is not also welcome because it cannot give a sense of how accurate prediction is. Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) is more often used in traffic prediction tasks. Given the above explanations, we consider to apply a multi-loss combine cross-entropy and weight MAPE together, and define the new loss function $\mathcal{L}$ as:

$$\mathcal{L} = w_{\text{heading}} \cdot \text{CE}(\text{heading}) + w_{\text{volume}} \cdot \text{MAPE}(\text{volume}) + w_{\text{speeding}} \cdot \text{MAPE}(\text{speeding}) \tag{1}$$

Here $\text{CE}(\cdot)$ represents the 2D cross-entropy loss with 5 classes $(0, NW, NE, SW, SE)$ while $\text{MAPE}(\cdot)$ is the 2D Mean Absolute Percentage Error.

Table 1. Average Predicted Values ($\bar{I}_{\text{predict}}$) verse its true value (I_{true}) for heading channel. We select (Moscow, 20181106, 258-260) as an example.

I_{true}	$\bar{I}_{\text{predict}}$
0	8.08
1	58.63
85	73.44
170	93.17
255	117.73

4.2 Multi-source Data Fusion and Online Learning

Multi-source data fusion is fashion and often be reported as useful in the research fields, but we did not embed any external information to the models for the final solution. We report that embedding external information *e.g.*, weather, holiday, and the day of week might lead to over-fitting. We tried to embed 20 external features into 64 dimension vector and add them to the output of the first 3×3 CONV of U-net, but the MSE error increased significantly when other features remain unchanged.

Here we propose a method that could apply external information to our model. For each predicted day, the model will be fine-tuned one more epoch using similar data of the predicted day. The similarities between data are measured by time, weather, and holiday features. If we just use the time to rank the similarity, then this method downgrade to an online learning method - the model tries to optimize recent inputs instead of global ones. By using this method, we had a deduction of 10^{-4} to loss.

4.3 Spatio-Temporal Ensemble Method

The motivation of spatio-temporal ensemble method stems from the large-scale spatio-temporal predictions performed in many online taxi-hailing service providers, *e.g.,* Uber, Didi, and Lyft. These predictions are usually completed by different teams from different domains, with different models. However, every single model may have some inherent defects due to insufficient information, and hence it is rational to ensemble the results of these models to improve the overall performance. It remains a challenging but essential issue to make an effective combination of multiple base models while leveraging the spatio-temporal information. To the best of our knowledge, the research in this vein is very sparse.

The prediction outcome can help the online car-hailing platforms accurately dispatch the fleet and lower the vehicle vacancy ratio, thus serving more passengers with their trip requests. Notably, even if the average prediction error reduction is small, compared with the optimal base model, the cumulative number of orders for a long period, say, one year, will be considerable in that the whole city can be seen as thousands of pixels, and DiDi and Uber run their business in hundreds of cities. Therefore, we design an attention-based deep ensemble net in dealing with the problems mentioned above[13].

4.4 Personalized Intelligent Transportation Prediction Systems

In our recent study, we managed to introduce the concept of personalization in the recommendation system to the prediction problem in intelligent transportation systems, and a more intelligent urban transportation system, which conforms to the individual preferences and is appropriate for regional traffic conditions, can thus be designed. Therefore, we explore personalization in the application of intelligent transportation systems. Aiming at the two common personalized demands in intelligent transportation systems, we define the two types of personalization in intelligent transportation systems on the basis of different application scenarios: User Personalization and Geographic Personalization[14]. Interdisciplinary knowledge and experience fusion is urgently needed classical traffic research. In addition to the construction of personalized intelligent transportation systems, there is also great potential for adversarial examples and transfer learning.

References

1. Ke, J., Zheng, H., Yang, H. & Chen, X. M. 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).
2. Yao, H. *et al.* Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).
3. Liu, Y., Liu, Z. & Jia, R. Deeppf: A deep learning based architecture for metro passenger flow prediction. *Transp. Res. Part C: Emerg. Technol.* **101**, 18–34 (2019).
4. Fu, R., Zhang, Z. & Li, L. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 324–328 (IEEE, 2016).
5. Zhang, J., Zheng, Y. & Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).
6. Geng, X. *et al.* Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *2019 AAAI Conference on Artificial Intelligence (AAAI'19)* (2019).
7. Li, Y., Yu, R., Shahabi, C. & Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations* (2018).
8. Pan, Z. *et al.* Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*, 1720–1730 (ACM, New York, NY, USA, 2019).
9. Yu, B., Yin, H. & Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)* (2018).
10. Koesdwiady, A., Soua, R. & Karray, F. Improving traffic flow prediction with weather information in connected cars: A deep learning approach. *IEEE Transactions on Veh. Technol.* **65**, 9508–9517 (2016).
11. Liao, B. *et al.* Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 537–546 (ACM, 2018).
12. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241 (Springer, 2015).
13. Liu, Y., Liu, Z., Lyu, C. & Ye, J. Attention-based deep ensemble net for large-scale online taxi-hailing demand prediction. *IEEE Transactions on Intell. Transp. Syst.* in press, DOI: 10.1109/TITS.2019.2947145 (2019).
14. Liu, Y., Lyu, C. & Liu, Z. Exploring personalization in the application of intelligent transportation systems. In *1st ACM SIGSPATIAL International Workshop on Ride-hailing Algorithms, Applications, and Systems (RAAS'19)*, in press, DOI: 10.1145/3357140.3365494 (ACM, Chicago, IL, USA, 2019).