**DeepCorNet: A efficient taxi-hailing prediction model**

**Feng Mao*, Zhiheng Li, Kai Zhang***

Tsinghua Shenzen Internation Graduate School, Tsinghua University, Shenzhen 518055, China

zhangkai@sz.tsinghua.edu.cn

*mf19@mails.tsinghua.edu.cn

**Abstract.** To enhance the taxi service, it is critical to improve the accuracy and efficiency of taxi-hailing prediction. Recently, the deep learning models have become the most advanced methods for taxi-hailing prediction, because they can well characterize the spatiotemporal dependencies of traffic states and achieve high prediction accuracy performances. However, most deep learning models suffer from expensive training cost. In this paper, we selected CNN as base model, the main challenge is how to convert taxi-hailing demand into image for most cases. To address this challenge, we proposed the DeepCorNet model, which generated the image of taxi-hailing demand by mutual information between regions. The experiment results showed that when taking both the model accuracy and training efficiency into consideration, the proposed model is probably more preferable, because it can conduct a relatively high prediction accuracy (close to GNN model) with much less computation efforts.

1. **Introduction**

Taxi-haling plays an important role in urban transportation system. However, these exists a paradoxical situation in taxi service. On one hand, taxi drivers often empty carrying to search for passengers. On the other hand, it is inconvenient for passengers to take a cab. It implies that taxi service run with low efficiency, which leads to environment pollution and traffic congestion [1]. Intuitively, taxi service enhancements have a direct impact on taxi drivers’ income, passengers travel comfort and traffic operation efficiency. In order to enhance the efficiency of taxi service, it is needed to improve the accuracy and efficiency of taxi-hailing prediction. This problem can be solved as a traffic prediction problem.

Existing traffic prediction approaches can be classified into two categories according to information sources: temporal feature-based methods and spatiotemporal feature based methods. The temporal feature-based methods include auto regressive integrated moving average (ARIMA) model [2], support vector regression (SVR) model [3], artificial neural networks (ANNs) [4] and Bayesian network [5]. However, this kind of methods do not utilize spatial information and have a low prediction accuracy.

To improve the prediction accuracy, numerous methods incorporate both the spatial and temporal information for traffic forecasting. These spatiotemporal methods can be roughly divided into two categories: classical statistic methods and machine learning methods. Some common classical statistic methods are multivariate time series method [6] and tensor-based method [7]. However, the former suffered from handing big data with high variance [8]. The tensor-based method is adapted to traffic forecasting task on small-scale traffic networks. The machine learning methods perform well on feature extraction and non-linear fitting, especially the deep learning models.
There are three main technical lines to apply the deep learning-based methods for traffic forecasting. The first one is Convolutional neural network (CNN)-based model [9-10]. The CNN algorithm performs well in characterizing the local dependencies of traffic states and is less sensitive to noise. The second one is Long and Short-Term Memory (LSTM)-based model [11-12]. The LSTM method is skilled in learning from the time series data with relatively long-time dependencies. The last one is Graph Neural Network (GNN)-based model [13-14], which can achieve high prediction accuracy for graph-structured data. However, the CNN-based model is commonly used for Euclidean data. The LSTM-based model suffers from time-consuming iterations and complex gate mechanisms. And the GNN-based model also requires long time to train the model.

To conduct an efficient and accurate taxi-hailing prediction, we selected the CNN method as base model. The main challenge lies in how to convert the origin taxi-hailing demand into Euclidean data (image). In this paper, we proposed DeepCorNet model to predict taxi-hailing demand. The image is generated by the mutual information between the demand series of regions. In the image, the neighboring pixels indicate two regions have strong correlations, and different channels represent the temporal information.

2. DeepCorNet Model

2.1. Problem description

For taxi-hailing prediction problems, we aim at estimating the taxi demand for different regions in the future time intervals based on the lagged taxi demand data. We define the problem as follow.

At time interval \( t \), suppose we have history data of taxi demand for \( M \) regions, \( \left[ X_1, X_2, \cdots, X_M \right] \), where \( X_i = \left[ x_i^{t-(N-1)}, x_i^{t-(N-2)}, \cdots, x_i^t \right] \) is the taxi-hailing demand for region \( i \) in the lagged \( N \) time interval. The task is to predict the taxi-hailing \( \hat{Y}_{t+1} = \left[ \hat{y}_1^{t+1}, \hat{y}_2^{t+1}, \cdots, \hat{y}_M^{t+1} \right] \) in time interval \( t+1 \).

2.2. Converting taxi-hailing demand into images

To convert taxi-hailing demand into image, an intuitive method is to partition the traffic network into a grid-based map [15]. This method is based on the assumption that there is possible causal relations between neighboring regions. However, the region can only be partitioned into rectangle in this kind of approach. For some cases, like vehicle dispatch, the region may be divided into other shapes, such as honeycomb structure and other irregular shapes.

In this paper, we aim to provide an architecture that can converting taxi-hailing demand into images for most cases. Recent research [16] pointed out that a strongly correlated source of information often indicated a causal relationship in traffic prediction. Therefore, we use statistical correlations between regions to model the spatial dependency and re-build the regions to an image. Besides, inspired by the work [17], we selected the mutual information coefficient (MI) to extract the spatial and temporal features with high correlation. The study [17] reported that the mutual information can capture non-linear statistical dependencies between variables.

Suppose the taxi-hailing demand for region \( i \) is denoted as \( D_i = \left[ x_i^1, x_i^2, \cdots, x_i^P \right] \), where \( P \) is the number of total time intervals for one week. The mutual information between region \( i \) and region \( j \) is calculated as follows:

\[
MI(D_i, D_j) = \sum_{x_i, x_j} p(x_i, x_j) \log \frac{p(x_i | x_j)}{p(x_i)}
\]

where \( p(x_i, x_j) \) is the joint probability distribution for variable \( D_i \) and \( D_j \), \( p(x_i | x_j) \) is the conditional probability distribution for \( D_i \) given \( D_j \), and \( p(x_i) \) is the marginal distribution for \( D_i \).
In this paper, we can obtain the $MI$ between two variables by wrapped function in python, thus we omit introducing the computing process of each parameters in Eq. (1).

We first map all the regions to a $\lceil \sqrt{M} \rceil \times \lceil \sqrt{M} \rceil$ square image $I$, where $\lceil \cdot \rceil$ is the ceiling function, the remaining $\lceil \sqrt{M} \rceil \times \lceil \sqrt{M} \rceil - M$ pixels of the image are padded with zeros. We denote the pixel located at $(i, j)$ of the image $I$ as $I(i, j)$, and its $\alpha$-neighbors as $A(i, j) = \{A_{i, j}^1, A_{i, j}^2, \ldots, A_{i, j}^\alpha\}$, where $A_{i, j}^k = \{\alpha_{i, j}^1, \alpha_{i, j}^k, \ldots, \alpha_{i, j}^{n_k}\}$ represent the k-neighbors of $I(i, j)$ and $n_k$ is the number of adjacent regions. We define the score of pixel $I(i, j)$ as

$$s(i, j) = \frac{1}{G} \sum_{k=1}^{G} s_{i, j}^k$$

$$s_{i, j}^k = \frac{1}{n_k} \sum_{k=1}^{n_k} MI(I(i, j), \alpha_{i, j}^{n_k})$$

Then, the total score of image $I$ is

$$s = \sum_{i=1}^{\lceil \sqrt{M} \rceil} \sum_{j=1}^{\lceil \sqrt{M} \rceil} s(i, j)$$

To make the pixel of strongly correlated regions be close, we arrange the order of the regions in the image to maximize the total score. However, this is an NP-hard problem, we implemented the method proposed in reference [16]. For each pixel, we look for its swap with other pixels that can achieve the maximum total score after swapping. This process will be repeated until there is no swap can further improve the total score. Finally, we can obtain a re-build region image $I_r$. Algorithm 1 presents the details of this process.

**Algorithm 1 Re-build the order of regions in the image**

**Input:** Input image $I$

1. Compute the total score $S_{old}$ of image $I$

   for all pixels $I(i_1, j_1)$ do
   while TRUE do
   for all pixels $I(i_2, j_2)$ excepting $I(i_1, j_1)$ do
   2. computer the new score $s_{new}(i_2, j_2)$ if swap $I(i_1, j_1)$ and $I(i_2, j_2)$ and save it in $s_{new}$
   if max($s_{new}$) - $s_{old} > 10^{-5}$ then
   3. $(i_3, j_3) = \text{arg max}(s_{new})$
   4. swap $I(i_1, j_1)$ and $I(i_3, j_3)$
   5. update $s_{old} = \text{max}(s_{new})$
   else
   break
   end for
   end for
1. Output: The re-build region image $I_r$

2.3. DeepCorNet model
The architecture of DeepCorNet model is illustrated in Fig. 1. Firstly, we convert the origin taxi-hailing demand data in the past $N$ time interval (shape: $N \times M$) into an $N$-channel image (shape: $N \times \sqrt{M} \times \sqrt{M}$) using the method introduced in section 2.2. Then the image is fed into a stack of convolutional layers to capture the spatial dependency. The obtained feature map from the top Conv1 is fed into $1 \times 1$ Conv2 layer, which can simultaneously characterize the temporal features. Finally, we can obtain taxi-hailing demand in time interval $t+1$ in output layer.

![Fig. 1 Architecture of DeepCorNet](image)

### 3. Experiments and Results

#### 3.1. Data description and processing

To evaluate the performance of the DeepCorNet, an empirical study of taxi passenger boarding demand prediction in Guangzhou is presented. In our experiment, we used the taxi GPS data of Guangzhou generated from Jun 1st to June 24th, 2015, with a total number of over 600 million effective logs, each of which comprised the information of taxi ID, GPS log time, GPS log position (the current longitude and latitude), taxi speed, carry status (carry passengers or not) and so on. We abandoned abnormal taxi trip data with distances less than 1 km. In this study, we used the weekday taxi GPS data of Guangzhou generated from Jun 1st to June 21th, 2015 for the experiments. The data from June 1st to June 5th and June 8th to 12th was selected as the training set, of which 80% was selected randomly to create the training set, and the remaining was used for validation. And the data from June 15th to June 19th was used to test the model.

Four prediction scenarios were set to test the performance of the proposed method:

1. Scenario 1: 10-min time interval, 0-10 min taxi passenger boarding demand prediction;
2. Scenario 2: 10-min time interval, 10-20 min taxi passenger boarding demand prediction;
3. Scenario 3: 15-min time interval, 0-15 min taxi passenger boarding demand prediction;
4. Scenario 4: 15-min time interval, 15-30 min taxi passenger boarding demand prediction.

In this paper, we used a square to divide an area into small regions. We implemented the strategy employed by Veloso [18], which use 500m×500m square to divide an area into small regions. By using the method, we can divide the test area of Guangzhou into 983 small regions and select 200 regions of the 983 regions for experiments.

#### 3.2. Performance index

To evaluate the effectiveness of the proposed prediction model, we focus on the following two error indexes, which are the mean absolute error (MAE), the mean square error (MSE).

\[
MAE = \frac{1}{M} \sum_{i=1}^{M} |y_{i}^{t+1} - \hat{y}_{i}^{t+1}|
\]  
\[
MSE = \frac{1}{M} \sum_{i=1}^{M} (y_{i}^{t+1} - \hat{y}_{i}^{t+1})^2
\]
3.3. Baselines and parameters setting

To evaluate the performance of the proposed model, multiple prevailing methods are selected for comparison.

**TGCN** is a GNN-based model, which combines the GCN and the GRU algorithm [19]. Many graph-base traffic prediction models selected this model as baseline for comparison.

**CNN** works well in extracting the local dependencies of traffic network, it takes spatiotemporal traffic state matrix of the whole network as input.

**LSTM** is skilled in learning temporal features and has been popular in time-series data prediction.

For the image re-build algorithm, α is set to be 3 to rearrange the regions orders.

For DeepCorNet, we used Scenario 1 to tune up parameters and conducted a grid search to determine the optimal parameters. We chose the hidden layer size from \{2, 3, 4, 5\}, the number of hidden units from \{32, 64, 128\}, filter size of Conv1 form \{3\times3, 4\times4, 5\times5\} and time lag from \{4, 6, 8\}. Finally, we obtained the best architecture of DeepCorNet: the hidden layer was 4, the number of hidden units was 128, the filter size of Conv1 and the time lag were 3\times3 and 6, respectively.

The structure and hyper-parameters of other models were determined based on the three-fold cross validation. Specifically, the hidden units of the TGCN model was 64, others hyper parameters of this model were identified based on the shape of adjacent matrix and input features. The structure of LSTM included: one input layer, two LSTM layer, two hidden layer and one output layer. The LSTM cell was \{64, 64\} and the hidden units for hidden layer were \{128, 128\}. The CNN structure contained two convolution layers with filter whose dimensions are (16, 3, 3), (32, 3, 3) respectively, and fully-connected layer with 128 hidden units.

3.4. Results and comparison

The prediction results on the test data sets with the eight methods were presented in Table 1 and Table 2. The results showed that DeepCorNet outperformed other methods except for TGCN model. Specifically, the TGCN model could achieve the best forecasting accuracy under all the four prediction scenarios, because the TGCN model possess strong ability to characterize the spatiotemporal dependencies of traffic networks. The prediction accuracy improvements from the DeepCorNet to other two models are obvious under all four scenarios. The possible reasons were that, for LSTM, it cannot well capture the spatial information. For CNN, since it partitioned the regions by positions, some adjacent regions may be less correlative. Furthermore, it validated that re-build the regions by mutual information can improve taxi-hailing prediction accuracy.

In this study, all the deep learning models are implemented on Tensorflow and experiments are performed by a PC Server (Intel(R) Xeon(R) CPU E5-2630 2.4GHZ, memory 128 GB). Figure 8 and Figure 9 presented the training epoch and training time of above eight models. Our proposed model yielded superior training efficiency except for CNN model, while LSTM are most difficult to learn. The efficiency improvement for DeepCorNet over other two models are large. For LSTM, it had complex gate-mechanisms, which led to enormous number of hyper parameters. Thus, the LSTM-based model suffers from expensive training cost. Under this circumstance, although the training epochs of LSTM was around 10, but it needed more than 1 min to run an epoch. The TGCN had the similar situations, because this model comprised of GCN and GRU networks.

Therefore, when taking both the model accuracy and training efficiency into consideration, the proposed DeepCorNet is probably more preferable, because it can conduct a relatively high prediction accuracy (close to TGCN) with much less computation efforts.

| Model     | [0-10min] prediction | [10-20min] prediction |
|-----------|----------------------|-----------------------|
|           | MAE      | MSE       | MAE      | MSE       |
| DeepCorNet| 2.801    | 15.050*   | 2.935    | 16.862    |
| TGCN      | 2.774*   | 15.229    | 2.929*   | 16.723*   |
| LSTM      | 2.932    | 17.135    | 3.192    | 20.058    |
| CNN       | 3.119    | 18.658    | 3.248    | 20.922    |
Table 2 Performance comparison of different methods with 15min time interval

| Model | [0-15min] prediction | [15-30min] prediction |
|-------|----------------------|-----------------------|
|       | MAE      | MSE     | MAE      | MSE     |
| DeepCorNet      | 3.676    | 27.222  | 3.926    | 32.001  |
| TGCN            | 3.491*   | 26.486* | 3.904*   | 31.940* |
| LSTM            | 3.821    | 28.232  | 4.215    | 35.754  |
| CNN             | 4.072    | 32.973  | 4.360    | 38.891  |

Fig.2 Training epoch of different models under four prediction scenarios

Fig.3 Training time of different models under four prediction scenarios

4. Conclusion

To improve the taxi service, it is vital to conduct accurate and efficient taxi-hailing prediction. In this paper, we proposed DeepCorNet method to balance the model complexity and accuracy. This approach selected CNN as base model. However, CNN is commonly used for Euclidean data. To address this problem, the proposed model generated the image by mutual information between two regions. We evaluated the performance of the proposed method with the multiple prevailing methods, and results showed that, when taking both the model accuracy and training efficiency into consideration, the proposed DeepCorNet is probably more preferable, because it can conduct a relatively high prediction accuracy (close to TGCN) with much less computation efforts. The proposed method in this paper would be useful to enhance taxi service and help taxis to become competitive with newer app-based modes.

An inevitable issue for taxi GPS data is data missing. In future work, we will explore the prediction performance under different degrees of data loss. Besides, we will investigate tensor-based methods which incorporates more spatiotemporal information for prediction.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No.2018YFB0105100) and by Science and Technology Innovation Committee of Shenzhen (No. JCYJ20170412172030008).

References

[1] Kong Xiangjie, Xia Feng, Wang Jinzhong, et al. Time-location-relationship combined service recommendation based on taxi trajectory data[J]. IEEE Transactions on Industrial Informatics, 2017, 13(3): 1202-1212.
[2] Kumar S V, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data[J]. European Transport Research Review, 2015, 7(3): 21.
[3] Ahn J Y, Ko E, Kim E Y. Predicting spatiotemporal traffic flow based on support vector regression and bayesian classifier[C]//2015 IEEE Fifth International Conference on Big Data and Cloud Computing. IEEE, 2015: 125-130.
[4] Chan K Y, Dillon T S, Singh J, et al. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 13(2): 644-654.
[5] Sun S, Zhang C, Yu G. A Bayesian network approach to traffic flow forecasting[J]. IEEE Transactions on intelligent transportation systems, 2006, 7(1): 124-132.
[6] Dunne S, Ghosh B. Regime-based short-term multivariate traffic condition forecasting algorithm[J]. Journal of Transportation Engineering, 2012, 138(4): 455-466.
[7] Ren Jiangtao, Xie Qiwei. Efficient od trip matrix prediction based on tensor decomposition[C]//2017 18th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2017: 180-185.
[8] Safikhani A, Kamga C, Mudigonda S, et al. Spatio-temporal modeling of yellow taxi demands in New York City using generalized STAR models[J]. International Journal of Forecasting, 2018.
[9] Yu Haiyang, Wu Zhihai, Wang Shuqin, et al. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks[J]. Sensors, 2017, 17(7): 1501.
[10] Ma Xiaolei, Dai Zhuang, He Zhengbing, et al. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction[J]. Sensors, 2017, 17(4): 818.
[11] Zhao Jianlong, Qu Hua, Zhao Jihong, et al. Towards traffic matrix prediction with LSTM recurrent neural networks[J]. Electronics Letters, 2018, 54(9): 566-568.
[12] Wang Jiawei, Chen Ruixiang, He Zhaocheng. Traffic speed prediction for urban transportation network: A path based deep learning approach[J]. Transportation Research Part C: Emerging Technologies, 2019, 100: 372-385.
[13] Yu Bing, Yin Haoteng, Zhu Zhanxing. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting[C]//Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18. 2018.
[14] Zheng Chuanpan, Fan Xiaoliang, Wang Cheng, et al. Gman: A graph multi-attention network for traffic prediction[C]//Proceedings of the AAAI Conference on Artificial Intelligence AAAI-17. 2017.
[15] Dai Xingyuan, Fu Rui, Zhao Enmin, et al. DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending[J]. Transportation Research Part C: Emerging Technologies, 2019, 103: 142-157.
[16] Hjelm R D, Fedorov A, Lavoie-Marchildon S, et al. Learning deep representations by mutual information estimation and maximization[J]. arXiv preprint arXiv:1808.06670, 2018.
[17] Veloso M, Phithakkitnukoon S, Bento C. Urban mobility study using taxi traces[C]//Proceedings of the 2011 international workshop on Trajectory data mining and analysis. 2011: 23-30.
[18] Zhao Ling, Song Yujiao, Zhang Chao, et al. T-gcn: A temporal graph convolutional network for traffic prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2019.