AGILE Association of Geographic Information Laboratories in Europe

Short-term Traffic Demand Prediction using Graph Convolutional Neural Networks

Aoyong Li[1][0000-0001-8032-5881] and Kay W. Axhausen[1][0000-0003-3331-1318]

Institute for Transport Planning and Systems, ETH Zurich, Switzerland
aoyong.li@ivt.bagu.ethz.ch

Abstract. Short-term traffic demand prediction is one of the crucial issues in intelligent transport systems, which has attracted attention from the taxi industry and Mobility-on-Demand systems. Accurate predictions enable operators to dispatch their vehicles in advance, satisfying both drivers and passengers. This study aims to predict traffic demand over the entire city based on the Graph convolutional network (GCNN). Specially, we divide the study area into several non-overlap sub-regions. Each sub-region is treated as a node, and a traffic demand graph is constructed. Then, we build three graph convolution networks based on three different weighted adjacency matrices, which represent three graph structures. Furthermore, a data-driven graph convolutional network (DDGCNN) is developed, which can capture the correlation between pairs of sub-regions automatically. Finally, we compare our models with other prediction methods, including three GCNNs with a normal adjacency matrix, an existing data-driven graph convolutional neural network, historical average, and random forest. Results show that the weighted adjacency matrix can improve the prediction performance compared with a normal adjacency matrix. In addition, we proved that our DDGCNN outperforms other predictors in three aspects, i.e., performance over the test set, performance over the time aspect, and the performance over the spatial aspect.

Keywords: Traffic demand prediction, Time-series forecasting, Graph convolutional neural network, Data-driven

1 Introduction

Mobility-on-Demand (MoD) systems, such as Uber, Didi, have gained great popularity all over the world. These systems are much convenient and flexible compared with traditional transport modes, such as subways and buses, by providing point-to-point travel with better comfort and convenience. In addition, they promote the sharing economy and enlarge the transportation capacities of the cities. One crucial operational challenge in MoD systems is the vehicle imbalances due to asymmetric vehicle demand. Vehicles might accumulate in some areas while being insufficient in others, which would impede the operations of the system. One approach to address this challenge is through surge pricing, which has been adopted by many MoD companies to attract drivers to the areas short of vehicles [1,13,21,22]. Another approach is through vehicle

AGILE: GIScience Series, 1, 2020.
Full paper Proceedings of the 23rd AGILE Conference on Geographic Information Science, 2020.
Editors: Panagiotis Partsinevelos, Phaedon Kyriakidis, and Marinos Kavouras
This contribution underwent peer review based on a full paper submission.
https://doi.org/10.5194/agile-giss-1-12-2020 | © Authors 2020. CC BY 4.0 License.

coordination, especially in the future Autonomous Mobility-on-Demand systems, where fleets of autonomous vehicles provide on-demand mobility services [3–5,14]. Although both approaches prove to be promising, it is widely shown that their performance (e.g., number of served requests, passenger waiting time, and profit) highly depends on the demand information [18]. Therefore, forecasting short-term demand throughout the city is of great importance to the systems.

Over the past few years, various data analysis models have been proposed to solve the short-term forecasting problem, including time-series analysis models [2,11], machine learning models [6,12], tree-based models [17]. Deep Learning (DL), a particular type of neural network, is a promising methodology, attracting much attention in the transport domain [19,20,23]. Among these DL models, a convolutional neural network (CNN) is defined to deal with regular grids, such as images. Since it can capture local spatial relations, CNN has been widely adopted in spatio-temporal prediction problems. Based on CNN, the whole city needs to be divided into a regular grid and is viewed as an image. However, the city generally has an irregular shape. In addition, traffic demand in a cell can be influenced by far-away areas rather than only the adjacent cells.

In this paper, we predict traffic demand throughout the city by improving graph convolutional neural network (GCNN). The study area is first divided into small non-overlap sub-regions, and the requests are aggregated based on a small interval (e.g., 15 minutes) in each sub-region. In this way, we get traffic demands series in each sub-region, which are used to train our models. The network input is the past traffic demands in all sub-region, the date information, while the outputs are the demand in future intervals. The GCNN based model is employed because: 1) it is used to deal with graph data, so the study area and sub-region can be in arbitrary shape; 2) it can capture the relation between any two subregions by defining different graph structure.

The improvements include two folds. First, we develop three GCNN models by employing a weighted adjacency matrix. The normal GCNN [8] defines the graph structure by using the adjacency matrix with only 0 and 1. 0 means no relation, while 1 denotes the relationship exists. However, in the real world, the influence between two sub-regions are different. For example, a close sub-region can provide more information compared with a distant one. Second, we develop a GCNN model with a data-driven adjacency matrix. Although the adjacency matrix can be easily defined, it is not trivial to define a suitable matrix because of different hidden correlations between sub-regions.

The rest of the paper is organized as follows. The literature review on traffic demand predictions is shown in Section 2. Section 3 shows the detail of our methodology. Section 4 introduces our data, experiments, and summarizes the main results. Section 5 concludes the paper and proposes the next steps.

2 Related work

Short-term travel demand prediction is to predict the number of requests at a certain period within a geographic area. Short-term here refers to a short time interval, which is usually less than one hour. This question is increasingly important to both transportation authorities and operators [6,19]. The emerging Mobility-on-Demand platforms, such as Uber, Lyft, and Didi, also have a strong motivation to obtain accurate forecasting for rider demands to improve their operational efficiency.

Various short-term demand prediction methods have been proposed, which can be categorized into parametric methods and nonparametric methods [9]. Parametric methods aim to use the historical data to fit a predetermined function linking the past and the present, where the most popular models are the Autoregressive Integrated Moving Average (ARIMA) models. Moreira-Matias et al. [11] proposed an ensemble framework consisting of three times-series analysis models, e.g., time-varying Poisson model, weighted time-varying Poisson model, and ARIMA (Autoregressive integrated moving average) model. The predicted demand was a weighted ensemble of predictions from three models. Instead of fixed ensemble weights, the weights were updated based on the prediction performances of previous time-steps. Davis et al. [2] shortlisted a couple of time-series techniques to fit the taxi data. In addition, they developed a multi-level clustering technique that can explore the correlation between adjacent subareas. These time-series methods are easy to be deployed, but they may not be able to capture the nonlinear relationship in traffic data.

Nonparametric methods, on the other hand, do not assume such a predetermined relationship, but rather attempt to identify historical data that is similar to the prediction instance. These methods can deal with nonlinear and non-stationary time series of traffic demand. Mukai and Yoden [12] predicted the taxi demand by a neural network, which adopted multi-features (e.g., demands in each region, day-of-week, and amount of precipitation) as the input data. Zhao et al. [24] implemented three predictors (i.e., Markov Predictor, LempelZiv-Welch Predictor, and Neural Network Predictor) to predict traffic demand at the building block level. In addition, the authors proved that the taxi demand is highly predictable by using entropy theory. Jiang et al. [6] predicted the demand for car-hailing service by using the least squares support machine (LS-SVM).

Deep learning (DL), a particular type of neural network, has been shown to be a reliable methodology that enables researchers to model complex nonlinear relationships between dependent variables and features. Several DL models have been applied to predict the traffic demand, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [19]. Among these models, CNNs exploit the local relation between grid cells, which have been widely adopted to capture spatial dependencies. Ke et al. [7] proposed a fusion convolutional long short-term memory network (FCL-Net), combining convolutional operations and LSTM layers to predict traffic demand. The explanatory variables include historical demands, travel time per unit travel distance, time-of-day, day-of-week, and weather conditions. To rank the importance of the explanatory variables, the authors employed a tailored spatially aggregated random forest. Yao et al. [20] proposed a Deep Multi-View Spatial-Temporal Network (DMVST-Net) framework to predict demand. In this framework, CNN is used to capture the features from the spatially nearby sub-regions.

However, CNN still has limitations. CNN is defined to deal with the data with regular grids, such as images, but the study area is usually in an irregular shape. Data preprocessing work is necessary to apply CNN in transportation [10]. A general approach is to fill the outside study area with zeros, which would, nevertheless, result in unnecessary storage. In addition, CNN only captures the local dependencies, but far-away areas could influence traffic demand in a cell. Graph convolutional network (GCNN) is an extension of CNN, which can deal with the data on an irregular domain [8]. By using different graph structures, GCNN can capture the relationship between different sub-regions. Therefore, we adopt GCNN to predict traffic demand.

3 Methodology

In this section, we first give some critical definitions related to the question. Then, we introduce the normal Graph convolutional neural network, in which we introduce how to construct the weighted adjacency matrix. Finally, a data-driven graph neural network is developed.

3.1 Problem Definition

Traffic request.
Traffic request describes when and where passengers need a vehicle, which is expressed as a tuple (*id,t, lng, lat*), where *id* is the identification of an order; *t* describes when passengers need the vehicle; *lng* and *lat* represent where passengers need a vehicle. The vehicle can be car-sharing vehicles, taxis, etc. In this study, without loss of generality, we use taxi data as an example in the experiment section.

Traffic demand.
Traffic demand represents the number of traffic requests in a sub-region at a specific time interval. For a study area, all the sub-regions are non-overlapping, which are denoted as ($l_1,...,l_N$). The time intervals are denoted as ($I_1,...,I_T$). The traffic demand in area *i* at interval *t* is denoted as $d_{i,t}$.

Short-term traffic demand prediction problem.
The demand prediction problem aims to predict the number of requests at a specific future period throughout the whole city, given the historical demands and some context features.

$$[d_{i,t+1},...,d_{i,t+T}] = f((d_{i,t-n},...,d_{i,t-1},d_{i,t}),X_{context}) \quad (1)$$

where *n* is the length of historical time series, *i* is the identification of subregion, $X_{context}$ represents context features, and *T* is the length of the time series needed to be predicted.

3.2 Graph Convolutional Neural Network

A graph convolutional neural network (GCNN) is a type of neural network that performs convolutional operations on graphs. We divide the whole city into several sub-regions and model the traffic demand as a graph $G=(V,E)$, where each vertex *i* represents a sub-region, characterized by a feature vector x_i, consisting of d_0 features. Each edge represents the connection between pairs of vertices. The graph structure is described as an $N \times N$ adjacency matrix *A*, whose elements *A*$_{ij}$ encode the connection degree between the signals at two vertices.

We aim to build a GCNN model to predict traffic demand for the next period. Suppose the GCNN model has several layers from the input to the output. For each layer *l*, the input is a feature vector $H^l = [x_i^l, i = 1 ... N]$, given as the previous layer *l-1,* where $x_i^l = \mathcal{R}^{N \times d^l}$ represents the features for each node *i* and d^l is the number of features for the current layer. Notice that the number of features per-vertex can be different in layers. The output vector H^{l+1} includes the traffic demand information in the next time period at each sub-region. Specifically, when describing a graph with adjacency matrix *A*, the layer-wise propagation rule can be defined as a nonlinear function:

$$H^{l+1} = f_A(H^l) \tag{2}$$

In this way, features for each node could become more abstract after each layer. Various types of GCNNs can be obtained by defining propagation rules in Eq. (2).

In this paper, we adopt a two-layer GCNN neural network and the following layer-wise propagation rule as [8]:

$$H^{l+1} = \sigma(D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} H^l W^l) \tag{3}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of a graph with added self-connections; I_N is the identity matrix; *D* is a diagonal matrix with $D_{ii} = \sum_j \tilde{A}_{ij}$; W^l is the trainable weight matrix; $\sigma(\cdot)$ is an activation function, such as ReLU, Tanh.

Normal adjacency matrix.

The adjacency matrix *A* represents the structure of the graph, which need to be defined in advance. The definition of such a matrix can be based on different rules [10]. This section proposes three data matrices to quantify the correlations between sub-regions.

Spatial distance matrix.

According to the first law of geography [16], near things are more related than distant things. A simple way to encoder the correlation is based on their spatial spherical distance, which is calculated by spherical distance (the shortest distance along with the Earth) in this study. Elements of Spatial distance matrix (SD) are then defined as:

$$SD_{ij} = Spherical\_dis\left(l_i, l_j\right) \tag{4}$$

where l_i and l_j are the central locations of sub-region *i* and sub-region *j*. Each entry of A_{SD} is set to *0* if $SD_{ij} \geq \kappa_{SD}$, otherwise $A_{SD_{ij}} = 1$.

Demand correlation matrix.

https://doi.org/10.5194/agile-giss-1-12-2020 |

The Demand Correlation matrix (DC) captures the correlations between sub-regions by employing historical demands. Each element is calculated by the Pearson Correlation Coefficient (PCC) based on the demand series between sub-region *i* and sub-region *j*.

$$DC_{ij} = PCC\left(h_i,\ h_j\right) \tag{5}$$

where h_i and h_j are the traffic demand series for sub-region *i* and sub-region *j*. A binary adjacency matrix is formed such that $A_{DC_{ij}} = 1$ if $DC_{ij} \geq \kappa_{DC}$, otherwise $A_{DC_{ij}} = 0$. κ_{DC} is a pre-defined threshold.

Demand euclidean matrix.

The demand euclidean matrix also measures the relationship by using historical demands. Instead of using PCC, we use mean euclidean distance to represent the absolute closeness between pairs of sub-regions.

$$DE_{ij} = \sqrt{\frac{\sum_{t=1}^{T}(h_{it} - h_{jt})^2}{T}} \tag{6}$$

where *T* is the length of the demand series. $A_{DE_{ij}}$ is set to 1 if $DE_{ij} \geq \kappa_{DE}$, otherwise $A_{DE_{ij}} = 0$.

Weighted adjacency matrix.

The above mentioned three adjacency matrices contain only zeros and ones, which only represent whether a relationship exists between two sub-regions, and the related sub-regions have the same effects for the target sub-region. However, in the real-world, a sub-region is related to various sub-regions in different weights. For example, for a given sub-region *i*, the adjacent sub-region *j* should have a more substantial influence compared with a farther sub-region. Therefore, we build three weighted adjacency matrix based on the above three rules.

For the DC matrix, a higher value means a better relationship. A weighted adjacency matrix is simply built as:

$$A_{WDC_{ij}} = \begin{cases} 0, & \text{if } DC_{ij} \leq \kappa_{DC} \\ DC_{ij} & \text{otherwise} \end{cases} \tag{7}$$

However, for SD and DE matrix, since a small number represents a good relationship, we build the corresponding adjacency matrix by using the reciprocal. To avoid zero denominators, we first replace zero elements with the minimum value of the remaining non-zero elements. Then, the two weighted adjacency matrices are denoted as:

$$A_{WSC_{ij}} = \begin{cases} 0, & \text{if } SD_{ij} \le \kappa_{SD} \\ \dfrac{1}{SD_{ij}^2} & \text{otherwise} \end{cases} \tag{8}$$

$$A_{WDE_{ij}} = \begin{cases} 0, & \text{if } DE_{ij} \le \kappa_{DE} \\ \dfrac{1}{DE_{ij}^2} & \text{otherwise} \end{cases} \tag{9}$$

3.3 GCNN with data-driven adjacency matrix

The hidden correlations between the two sub-regions are heterogeneous. Thus, the predefinition of the adjacency matrix A is not trivial. Hence, it is difficult to encode them using just a matrix such as the SD, DC, and DE matrix. The intuitive idea is to learn the filter from the data. Let $\hat{A} = D^{-\frac{1}{2}}\tilde{A}D^{-\frac{1}{2}}$, Lin et al. [10] replaced Equ. (3) as:

$$H^{l+1} = \sigma(\hat{A}H^{l}W^{l}) \tag{10}$$

However, Equ. (10) has restrictions for $\hat{A}$. 1) It considers $\hat{A}$ as a symmetrical matrix, which means pairs of sub-regions have the same and mutual influence. 2) $\hat{A}$ only has positive elements, which only allow the positive influence between pairs of sub-regions. In this paper, we use a similar form as,

$$H^{l+1} = \sigma(\check{A}H^{l}W^{l}) \tag{11}$$

where $\check{A}$ is an adjacency-like matrix, which is learned from data as W_l. Each entry in $\check{A}$ could be negative, positive, and zero.

4 Experiments and results

4.1 Experiment setup

The data utilized in this study are from TIL between June 2015 to June 2016 [15]. Each record includes pickup time, dropoff time, pickup longitude, pickup latitude, dropoff longitude, dropoff latitude. Since over 80% of these records have origins and destinations falling within the Manhattan island, we only focus on taxi data within Manhattan island for demand prediction. This study area is furthermore divided into grid uniformly, where each cell refers to a sub-region. The edge of the cell if 500 meters. In addition, these pickups are aggregated in a 15 minutes time interval. Furthermore, we only consider only regions where at least has one request per day on average had been made. After eliminating the regions that do not satisfy our limit, 269 regions were left.

In the experiment, the first 11 months' of the data sets (between June 2015 to April 2016) are used as the training set, the following month (May 2016) is used as the

Full paper Proceedings of the 23rd AGILE Conference on Geographic Information Science, 2020.
Editors: Panagiotis Partsinevelos, Phaedon Kyriakidis, and Marinos Kavouras
This contribution underwent peer review based on a full paper submission.
https://doi.org/10.5194/agile-giss-1-12-2020 | © Authors 2020. CC BY 4.0 License.

validation set, and the last month (June 2016) is set as the test set. The DC and DE matrices are built from the training set. The SD is calculated based on the center point of sub-regions.

To consider the historical demands, we select the past eight intervals' demands as features for each sub-region, which are normalized to [0, 1]. In addition, the features also include *day of week* and *hour*, which are categorical variables and encoded using Embedding methods. Based on section 3.2, we build six types of GCNN models based on the known adjacency matrix, which are labeled as GSdEa, GPdEa, GEdEa, GSdUa, GPdUa, GEdUa, where Sd, Pd, and Ed means spatial distance, demand correlation, and demand euclidean based adjacency matrix respectively. Ea and Ua denote normal adjacency and weighted adjacency matrix, respectively. In addition, we build our data-driven GCNN, according to section 3.3, which is denoted as DSu. We also select another data-driven GCNN for comparison, which is based on [10], denoted as DSp.

4.2 Performance Metrics

Here, we use Symmetric Mean Absolute Percentage Error (sMAPE) and Root Mean Square Error (RMSE) to examine the performance of GCNN and other comparison methods [19]. From the statistic perspective, RMSE shows the absolute difference between the predicted value and the real value, while the sMAPE describes a percentile error. Both metrics are expressed as

$$sMAPE = \frac{1}{M}\sum_{i=1}^{M}\frac{|d_i - d_i|}{d_i + d_i + c} \tag{12}$$

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(d_i - d_i)^2} \tag{13}$$

where d_i is the real demand, d_i is the predicted demand, and M is the length of individual demand. c in sMAPE is a small number (here, we make c=1), which is to avoid zero in the denominator. By using a different set of individual demands, we can get the prediction performance in various sub-regions and periods.

4.3 Performance between GCNN-based models

In this section, we compare different GCNN-based models, i.e., six GCNN models with a given adjacency matrix and two data-driven GCNN models. The results are listed in Table 1. The RMSE and sMAPE are calculated for the same test set. In the first block, the first three rows show the results of GCNN models with normal adjacency matrices, followed by the corresponding GCNN models with given weighted adjacency matrices. The second block gives results for two data driven GCNN models.

We can also get some other conclusions. 1) The first block shows GSdUa, GPdUa, and GEdUa get a lower RMSE than GSdEa, GPdEa, and GEdEa, respectively. In addition, although GPdEa and GPdUa get a similar sMAPE. GSdUa and GPdUa get a much

https://doi.org/10.5194/agile-giss-1-12-2020 |

better performance in terms of sMAPE. It means that the weighted adjacency matrices can improve the performance of normal adjacency matrices. 2) The improvement of the weighted adjacency matrix depends on how to define the relationship between sub-regions. 3) Although weighted adjacency can enable GCNN to get a better a good performance, DSp model obtains the best performance under RMSE and sMAPE.

Table 1. Comparison between GCNN-based models on test set

Model	RMSE	sMAPE
GSdEa	23.549	0.420
GPdEa	12.970	0.203
GEdEa	23.538	0.419
GSdUa	**6.765**	0.169
GPdUa	12.813	0.212
GEdUa	6.814	**0.158**
DSp	6.167	0.153
DSu	**5.880**	**0.143**

4.4 Methods for Comparison.

In addition, we also compare GCNN models with another two prediction models, the Historical Average (HV) and Random Forest (RF).

Historical Average.

This approach predicts future demand as the average value of the same period and the same sub-regions during the past weeks. For example, if it is 8:00 am on Monday, the predicted value would be the average of demands at 8:00 am in the past 5 Mondays.

Random Forest.

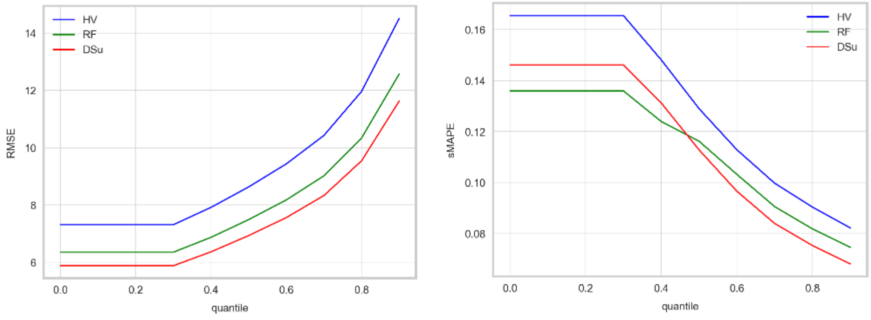
Random forest (RF) is a powerful ensemble learning model, which has been widely used in many regression and regression tasks. The method constructs a handful of decision trees at the training process, and output is the mean of predictions of all the individual trees. Compared to other ensemble strategies, the particularity of random forests lies in the process by which the trees are built. Moreover, the obtained trees are not pruned. These models can also be used for estimating variable importance. They are reasonably fast to obtain and can be easily parallelized if more speed is required.

We first report the prediction performance over the whole test set. Then, we show the prediction performance over the entire city. Finally, we show the performance at all the sub-regions.

Prediction performance Over test set.

In this section, the three models are trained using the same training set, and the inference results are based on the same test set. Instead of only showing the performance based on the whole test, we test the performance for the demand in various sizes, which is reported in Figure 1. The *x*-axis is quantile, which means only the elements that are higher the given quantile are in the test set. The two *y*-axises are RMSE or sMAPE. For example, if *x* is 0.5, we get the partial test set where all the elements are higher than 0.5 quantiles. Then, the RMSE and sMAPE are calculated based on the partial test set and the corresponding predicted values.

We can easily find that when real demand increases, the RMSE also becomes larger while sMAPE decreases. The DSu can always get the best performance under RMSE (Figure 1a). In terms of sMAPE (see Figure 1b), although DSu does not get the best performance for the whole data set, it gives a more accurate prediction when the demand is high. In fact, the demand intensity area needs much more attention to vehicle dispatching. Thus, predicting demand accurately in high demand set is more useful.

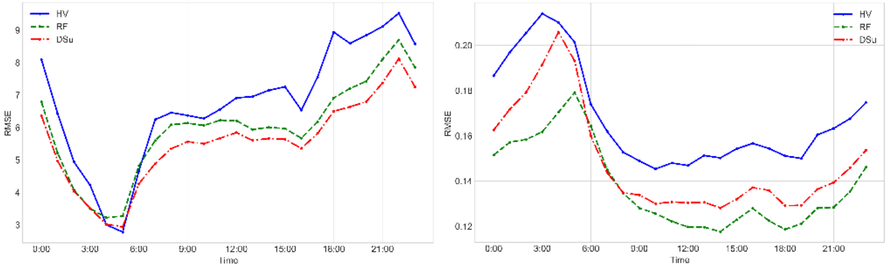


(a) RMSE (b) sMAPE

Fig. 1. Performance comparison over the test set

Prediction results Over the whole day.

Figure 2 reports the prediction performance over the entire city between the three predictors. All methods share some common patterns in both metrics. For instance, RMSE reaches its minimum values at about 4 am and maximum values at about 11 pm. sMAPE gets the largest value at about 3 am. In terms of RMSE, DSu gets the best performance at most times, but RF gets a better performance under sMAPE. One possible reason is that the test set contains more low demand value throughout the day.



(a) RMSE

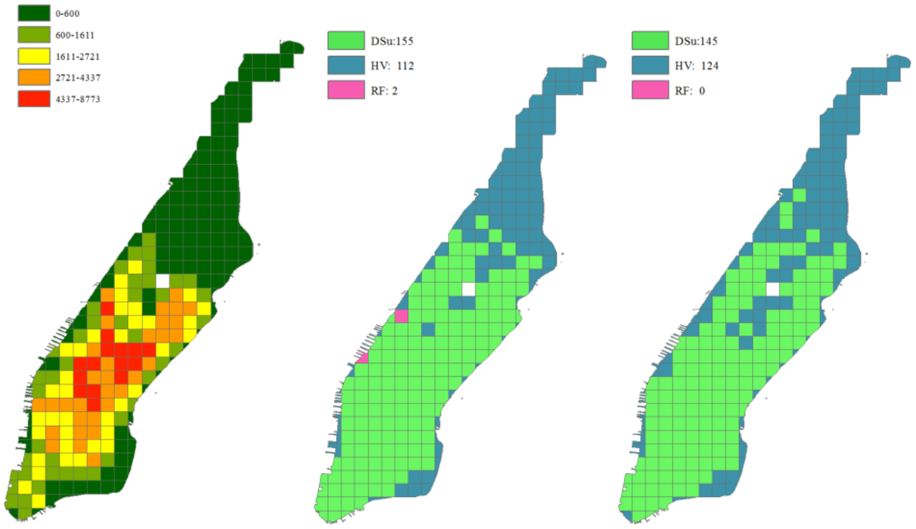
(b) sMAPE

Fig. 2. Performance comparison over the test set

Performance over the area.

Figure 3 reports the prediction performance over the space. In Figure 3a, each cell represents the average daily demand in the test set, where red color means high demand, and green color means low demand. The imbalance of traffic demand can be easily observed. The high demand cells are mainly distributed in the southwestern part of Manhattan. In Figure 3b and 3c, we represent each cell with a method that gets the best performance according to RMSE and sMAPE, respectively. The two sub-figures reflect that DSu gets the best performance in most sub-regions. In the 269 sub-regions, 155 sub-regions get the best performance under RMSE, and 145 sub-regions get the lowest sMAPE. In addition, most of the cells in DSu are mainly distributed in the high demand area. The historical average (HV) also gets the best performance in several sub-regions, but all of these cells are concentrated in the low demand area.

https://doi.org/10.5194/agile-giss-1-12-2020 | © Authors 2020. CC BY 4.0 License.



(a) Daily demand (b) RMSE (c) sMAPE

Fig. 3. Performance comparison over the test set

5 Conclusions

In this study, we propose a data-driven graph convolutional neural network (DDGCNN) for short-term traffic demand predicting. In addition, three weighted adjacency matrix is provided for the normal Graph convolutional neural network. By learning from historical demand patterns, the proposed models can make traffic demand the entire city. To evaluate the proposed models, the proposed models are compared with another four graph convolutional neural networks and two typical prediction methods by using NYC data. Several findings can be drawn based on the results. 1) The weighted adjacency matrix enables the GCNN to get better performance compared with the normal adjacency matrix. The degree of improvement differs in the construction of matrices. 2) The proposed model outperforms the other predictors, especially for high demand. 3) Our DDGCNN gets the best performance at a different time and most of the sub-regions.

Although our methods get excellent performance, they still suffer some limitations. The limitations and the possible improvements are summarized as follows. 1) In our method, we only consider historical demand information. It is not enough to describe all the influence factors. In future research, the work will consider more information to the input, such as weather, land use, and social events. 2) GCNN is good at learning and utilizing spatial dependencies. Time factors are also essential influence factors for traffic demand prediction. In the next step, the recurrent neural network is another type of deep learning model to deal with time-series data, which is worthy of fusing into the model to consider spatial dependence and time dependence simultaneously. 3) Based on our results, we can find that no one method can get the best performance in all the periods and sub-regions. It is worth to study how to create an ensemble model and get the best performance. 4) Finally, it is useful to study how to organize the vehicles in the entire city according to the predicted demand.

References

1. Battifarano, M., Qian, Z.: Predicting real-time surge pricing of ride-sourcing companies. Transportation Research Part C: Emerging Technologies 107, 444– 462 (2019).
2. Davis, N., Raina, G., Jagannathan, K.: Taxi demand forecasting: A hedge-based tessellation strategy for improved accuracy. IEEE Transactions on Intelligent Transportation Systems 19(11), 3686–3697 (2018).
3. Dia, H., Javanshour, F.: Autonomous shared mobility-on-demand: Melbourne pilot simulation study. Transportation Research Procedia 22, 285–296 (2017).
4. Hyland, M., Mahmassani, H.S.: Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transportation Research Part C: Emerging Technologies 92, 278–297 (2018).
5. Iglesias, R., Rossi, F., Zhang, R., Pavone, M.: A BCMP network approach to modeling and controlling autonomous mobility-on-demand systems. The International Journal of Robotics Research 38(2-3), 357–374 (2019).
6. Jiang, S., Chen, W., Li, Z., Yu, H.: Short-Term Demand Prediction Method for Online Car-Hailing Services Based on a Least Squares Support Vector Machine. IEEE Access 7, 11,882–11,891 (2019).
7. Ke, J., Zheng, H., Yang, H., Chen, X.: Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transportation Research Part C: Emerging Technologies 85, 591–608 (2017).
8. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. CoRR abs/1609.02907 (2016).
9. Li, A., Axhausen, K.W.: Comparison of short-term traffic demand prediction methods for transport services. Arbeitsberichte Verkehrs-und Raumplanung 1447 (2019)
10. Lin, L., He, Z., Peeta, S.: Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transportation Research Part C: Emerging Technologies 97, 258–276 (2018).
11. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: Predicting taxi–passenger demand using streaming data. IEEE Transactions on Intelligent Transportation Systems 14(3), 1393–1402 (2013).
12. Mukai, N., Yoden, N.: Taxi Demand Forecasting Based on Taxi Probe Data by Neural Network. Intelligent Interactive Multimedia: Systems and Services, pp. 589–597. Springer Berlin Heidelberg (2012).
13. Nourinejad, M., Ramezani, M.: Ride-sourcing modeling and pricing in nonequilibrium two-sided markets. Transportation Research Part B: Methodological (2019).
14. Salazar, M., Lanzetti, N., Rossi, F., Schiffer, M., Pavone, M.: Intermodal autonomous mobility-on-demand. IEEE Transactions on Intelligent Transportation Systems (2019).
15. TLC Trip Record Data. URL https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, last accessed 2020/06/06.
16. Tobler, W.R.: A computer movie simulating urban growth in the detroit region. Economic geography 46(sup1), 234–240 (1970).

17. Vanichrujee, U., Horanont, T., Pattara-atikom, W., Theeramunkong, T., Shinozaki, T.: Taxi demand prediction using ensemble model based on rnns and xgboost. In: 2018 International Conference on Embedded Systems and Intelligent Technology International Conference on Information and Communication Technology for Embedded Systems (ICESIT-ICICTES), pp. 1–6. IEEE (2018).
18. Wen, J., Nassir, N., Zhao, J.: Value of demand information in autonomous mobilityon-demand systems. Transportation Research Part A: Policy and Practice 121, 346–359 (2019).
19. Xu, J., Rahmatizadeh, R., Boloni, L., Turgut, D.: Real-time prediction of taxi demand using recurrent neural networks. IEEE Transactions on Intelligent Transportation Systems 19(8), 2572–2581 (2018).
20. Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z.: Deep multi-view spatial-temporal network for taxi demand prediction. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018).
21. Zha, L., Yin, Y., Du, Y.: Surge pricing and labor supply in the ride-sourcing market. Transportation Research Procedia 23, 2–21 (2017).
22. Zha, L., Yin, Y., Xu, Z.: Geometric matching and spatial pricing in ridesourcing markets. Transportation Research Part C: Emerging Technologies 92, 58–75 (2018).
23. 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).
24. Zhao, K., Khryashchev, D., Freire, J., Silva, C., Vo, H.: Predicting taxi demand at high spatial resolution: Approaching the limit of predictability. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 833–842 (2016).