Short-term transit ridership prediction

A comparative study among the graph convolutional network, the convolutional neural network, and linear regression

Kaihang Zhang* Siqi Feng†

Department of Civil and Environmental Engineering, University of California, Berkeley

Abstract

Ridership prediction is crucial for public transit authorities and companies, since proper prediction results would help traffic engineers reduce operating costs and improve the mobility of the entire transportation system. Various predictive methods have been developed and applied, ranging from theory-driven to data-driven models. In this paper, we aim to compare three models for ridership prediction on the station-level, namely Linear Regression (LR), the conventional Convolutional Neural Network (CNN), and the Graph Convolutional Network (GCN). We used the Bay Area Rapid Transit (BART) ridership data in the year of 2019 and 2020 to train and test the models. Through our empirical study, we find out that the LR model is unsuitable for ridership prediction, while the CNN model has fine results but also has the over-estimation problem which leads to lots of odd predictions. The GCN model takes the advantage of the spatial information as hyperdata. Therefore, it captures the ridership patterns very well and predicts the ridership closest to the ground truth with the lowest errors. We also compared the performance of each model in 2019 and 2020, which shows that the GCN model still performs the best after the COVID-19 pandemic hit the ridership significantly.

1 Introduction

During the COVID-19 pandemic, public transit ridership has been significantly influenced by the virus and the relevant policies such as lockdown. As the industry is recovering from the pandemic, ridership prediction is crucial for transit authorities for them to reduce the operating costs and to improve the mobility of the entire transportation system.

Ridership demand prediction is similar to traffic state (e.g., traffic flow) forecasting for transportation networks. It aims at predicting the number of passengers traveling from one station to another, in a specific future time horizon, based on historical trip data or other station features such as the land usage at the vicinity of one station, or demographic census data. Now that the

*Kaihang Zhang, SID: 3037432723; Email: kaihangzhang@berkeley.edu.

†Siqi Feng, SID: 3037432190; Email: siqi_feng@berkeley.edu.

task of ridership prediction is comparable with that of traffic state forecasting, the methods should also be transferable from the latter to the former, notwithstanding some existent literature being focused on ridership itself.

Various predictive methods have been proposed and applied so far, ranging from theory-driven to data-driven models. More specifically, linear regression (LR) is a representative for the former, while convolutional neural networks (CNNs) and the burgeoning graph convolutional networks (GCNs) are representatives for the latter.

As a novel machine learning approach, GCNs are able to capture both spatial and temporal features, hence they are suitable for short-term ridership prediction. In our case, transit ridership is highly dependent on road network connectivity, but is also susceptible to time of the day. With a GCN, we are capable of jointly considering transit connectivity and temporal properties at the vicinity of each transit station. Therefore, the GCN approach would hopefully outperform conventional models both in prediction and robustness.

The motivations for our study are twofold, with the methodological motivation described above. From the application aspect, GCNs have rarely been applied to real-world transit ridership prediction, let alone dealing with drastic declines caused by the pandemic.

We conducted a comparative study among graph convolutional networks (GCNs), convolutional neural networks (CNNs) and linear regression (LR) to realize ridership prediction on the station level. The ridership dataset of Bay Area Rapid Transit (BART) was used for model training and testing. The computational efficiency of the three methods is also compared.

The remainder of this paper is organized as follows. In Section 2, we briefly review the typical literature on traffic state prediction. In Section 3, we define the research problem mathematically. In Section 4, we introduce the three approaches applied in this paper. In Section 5, we analyze the BART ridership dataset. In Section 6, we present and discuss the results of these approaches. Finally, we provide our conclusions and opportunities for future work in Section 7.

2 Literature review

Numerous predictive methods have been investigated in the field of traffic state prediction, which can be categorized into two groups, namely parametric and non-parametric models (Zhao et al., 2020). Popular parametric models include Kalman filter based models (Okutani and Stephanedes , 1984), autoregressive integrated moving average models (ARIMA; Hamed et al., 1995; Lee and Fambro, 1999), multivariate regression (Clark, 2003), etc. However, these models are weak in capturing the complicated spatial-temporal features in transportation systems (Zhao et al., 2020), which calls for more advanced methods. Non-parametric models, such as machine learning models, are feasible solutions to such a prediction challenge. Recent research shows that these models have gained superiority in transportation prediction scenarios (Ke et al., 2017; Ma et al., 2019; Zhang et al., 2020).

Despite their fair performance in prediction, the construction of traffic data in a Euclidean way is questionable (Zhao et al., 2020), because transportation networks are de facto complex non-Euclidean graphs. To overcome this issue, researchers deployed graph convolutional networks (GCNs) into the transportation field. The idea of graph network was first introduced by (Bruna et al., 2014) and was extended to convolutional network by (Kipf and Welling, 2017). Because of the outstanding performance of GCNs for tackling complex non-Euclidean networks, there are also many applications in transportation engineering (Chen et al., 2021; Du et al., 2021; Jin et al., 2020; Lu et al., 2020; Wu et al., 2019; Yu et al., 2018; Zhao et al., 2020), most of which are combinatory models of GCN with a temporal convolutional network to incorporate spatio-temporal features. Notably, Jin et al. (2020) blended into the model a variational autoencoder that can automatically capture latent traffic features. Besides, external information such as meteorological information and network vertex correlation were also investigated. Chen et al. (2021) developed a parallel-structured deep learning model that includes a GCN for metro ridership prediction. Du et al. (2021) argued that traffic states could be greatly affected by weather conditions, and added an additional fully-connected layer to describe the weather influence. Liu et al. (2020) pointed out that local spatial dependency does not necessarily represent inter-station characteristics. Therefore, they took the similarity graph and the correlation graph into consideration to further incorporate the underlying station-wise interrelationship.

3 Problem definition

Ridership data for each transit station is typically aggregated by specific time intervals, resulting in a batch of time series data for further manipulation. In this study, we aim to develop models (or functions) that can predict the trip counts of each OD pair in next q time intervals given the trip counts in previous p time intervals. Thus, the input demand data of n OD pairs form a $n \times p$ matrix:

$$\boldsymbol{X}_{n\times p} = [\boldsymbol{x}_{t-p},\ \boldsymbol{x}_{t-p+1},\ \ldots,\ \boldsymbol{x}_{t-1}] = \begin{bmatrix} x^1_{t-p} & x^1_{t-p+1} & \cdots & x^1_{t-1} \\ x^2_{t-p} & x^2_{t-p+1} & \cdots & x^2_{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ x^n_{t-p} & x^n_{t-p+1} & \cdots & x^n_{t-1} \end{bmatrix} \tag{1}$$

where x^j_{t-i} denotes the demand of the jth OD pair in the ith time interval before timestamp t.

Then, the predicting problem can be formulated as learning a function f:

$$f : \boldsymbol{X}_{n\times p} \to \boldsymbol{X}_{n\times q} \tag{2}$$

where $\boldsymbol{X}_{n\times q} = [\boldsymbol{x}_t,\ \boldsymbol{x}_{t+1},\ \ldots,\ \boldsymbol{x}_{t+q-1}]$. We chose $p = 5$ and $q = 4$ for this study, i.e., we utilized the trip counts in 5 hours to predict the trip counts in the next 4 hours.

4 Methodology

4.1 Linear regression

Linear regression is one of the most straightforward and widely-used method for prediction based on time series data. The method multiplies the historical OD matrix with a matrix of coefficient estimates to predict future demands:

$$\hat{\boldsymbol{X}}_{n\times q} = \boldsymbol{X}_{n\times p}\,\hat{\boldsymbol{\beta}}_{p\times q} = \begin{bmatrix} x^1_{t-p} & x^1_{t-p+1} & \cdots & x^1_{t-1} \\ x^2_{t-p} & x^2_{t-p+1} & \cdots & x^2_{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ x^n_{t-p} & x^n_{t-p+1} & \cdots & x^n_{t-1} \end{bmatrix} \begin{bmatrix} \hat{\beta}_{11} & \hat{\beta}_{12} & \cdots & \hat{\beta}_{1q} \\ \hat{\beta}_{21} & \hat{\beta}_{22} & \cdots & \hat{\beta}_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\beta}_{p1} & \hat{\beta}_{p2} & \cdots & \hat{\beta}_{pq} \end{bmatrix} \tag{3}$$

4.2 Convolutional neural network

The CNN structure used in this paper is illustrated in Fig. 1. CNN is widely being applied in the filed of computer vision, where images are taken as the input. In our scenario, OD matrices are treated as an image and fed into CNN model. p OD matrices are stacked together before being inputted. The kernels shown in Fig. 1 as yellow squares are moving across OD matrices, dot-multiple the areas they covered, and result in one value in the hidden layer, shown as red spots. The operation that the moving kernels take the dot productions of themselves and the area they covered is considered as convolution. The learnable coefficients are which values are exactly in the kernel, and the coefficients between each layers. After the convolution process shown in Fig. 1, the output is finally q OD matrices presented as a 3-dimensional tensor.

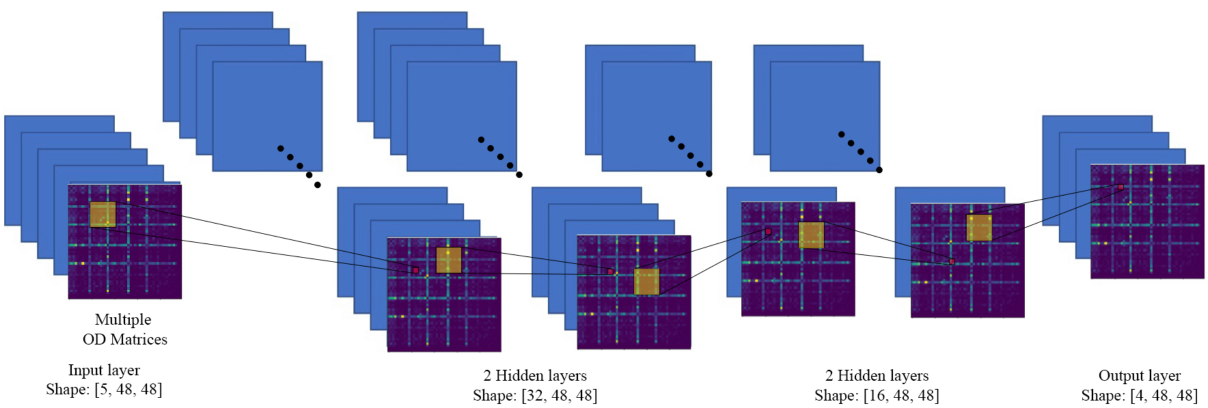


Figure 1: Illustration of a CNN

4.3 Graph convolutional network

The GCN structure used in this paper is illustrated in Fig. 2. Unlike CNN, which learn the kernel during the training process, GCN takes the adjacency matrix of stations as an input and operate convolution by utilizing it. As shown in Eqs. (4) and (5) (Kipf and Welling, 2017), where $\sigma(\cdot)$ is the activation function, $\boldsymbol{H}^{(l)}$ and $\boldsymbol{W}^{(l)}$ are the hidden state matrix and the weight matrix in the lth layer of the neural network, respectively, $\boldsymbol{A}$ is the connectivity matrix $\boldsymbol{C}$ plus an identity matrix, i.e., $\boldsymbol{A} = \boldsymbol{C} + \boldsymbol{I}$, and $\boldsymbol{D}$ is the diagonal degree matrix defined as $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$. We can obtain an initial insight from the degree matrix that it shows how many stations does each station is connected with. Eq. (5) could be considered as an normalization operation. The functionality of $\widetilde{\boldsymbol{A}}$ can be illustrated in Fig. 2, the flow from each origin station to the destination station is summed from the flow of its adjacent stations. This summation operation is considered as convolution in a GCN.

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right) \implies \boldsymbol{H}^{(l+1)} = \sigma\left(\widetilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right) \tag{4}$$

$$\widetilde{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \tag{5}$$

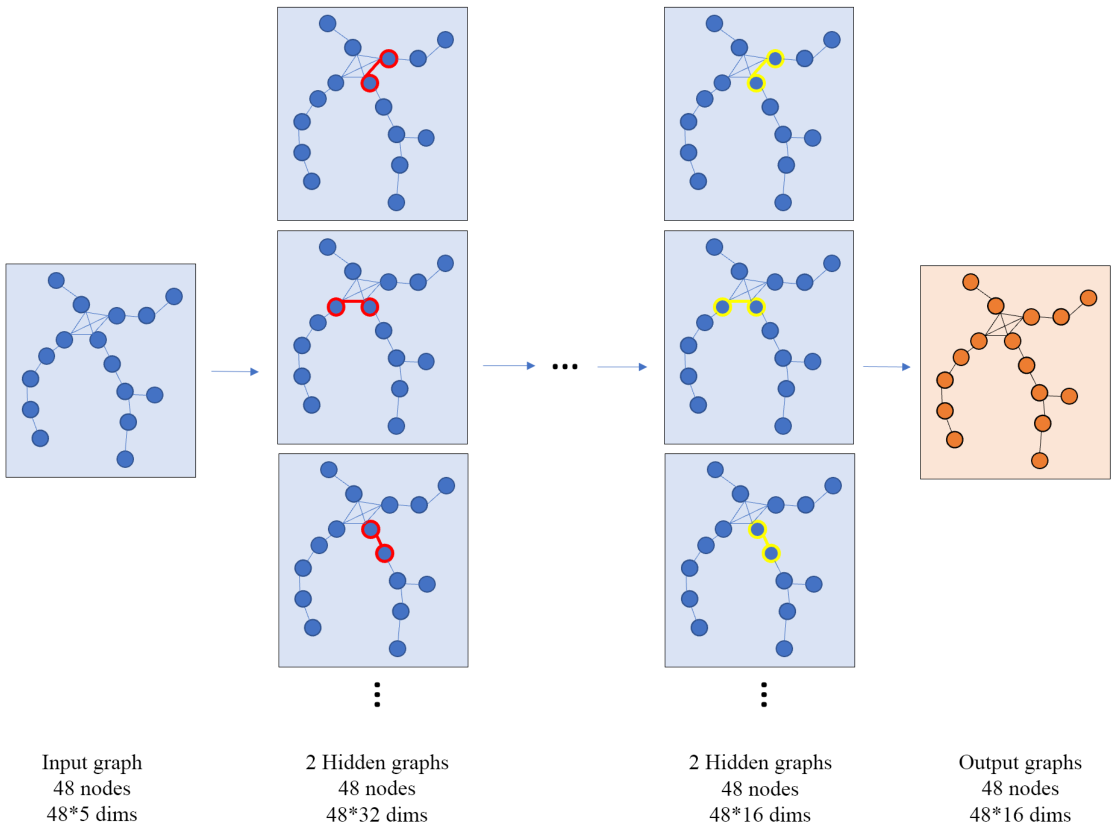


Figure 2: Illustration of a GCN

5 Data

5.1 BART ridership data

The BART ridership dataset provides us with the OD trip data in hour level and month level[1]. In this study, we used the hourly data collected in January 2019 and April 2020 because we intend to do short-term ridership prediction in the hour level. As shown in Table 1, the raw data (of any year) consists of five columns, which are date, hour, origin station, destination station, and trip counts[2]. The statistical information of the OD data is shown in Table 2.

[1] The data can be downloaded from https://www.bart.gov/about/reports/ridership.

[2] See https://api.bart.gov/docs/overview/abbrev.aspx for the full names of BART stations.

Table 1: BART OD Trip Data

Date	Hour	Origin	Dest.	Counts
1/1/19	0	12TH	12TH	3
1/1/19	0	12TH	16TH	4
1/1/19	0	12TH	ANTC	1
...	...	...	...	...

Table 2: Summary of the OD data

Year	Mean	Max.	Min.	Std. dev.
2019	5.42	1056	0	22.73
2020	925	9	0	9.62

We converted the 1-dimensional trip count array into a 3-dimensional OD tensor, the shape of which are origin stations, destination stations, and time of the year. For example, the OD tensor of the year 2019 has shape 48 by 48 by 8760 because there are 48 stations (the Milpitas and Berryessa are excluded because they did not open for service until June 2020) in total and there are $24 \times 365 = 8760$ hours in 2019 (see Fig. 3). For visualization, we summed up the trip counts across the year and obtained a 2-dimensional matrix, which is shown in Fig. 4, where we can find that the four stations with significantly higher ridership are all located in downtown San Francisco: CIVC (Civic Center), EMBR (Embarcadero), MONT (Montgomery St.), and POWL (Powell St.). For the GCN model, an adjacency matrix is an important data component and is visualized in Fig. 5 with purple representing the directly connected OD pairs, and yellow denoting the others.

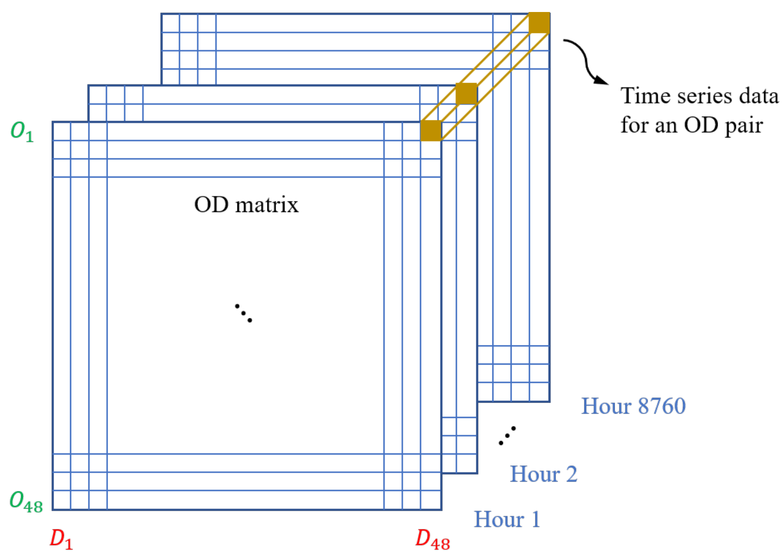


Figure 3: Illustration of time series data

Since the transit data is highly periodic, we trained our models upon a 1-month scheme instead of the entire year[3]. We visualized the daily trip counts from EMBR to DBRK (Downtown Berkeley) in Fig. 6 in order to illustrate, from a temporal aspect, the significant impact of pandemic on transit ridership. As shown in Fig. 6, we focus on the ridership prediction in January 2019 (a normal pre-pandemic month) and April 2020 (the first month after strict lockdown). We treated each month as 720 hours, trained our model by the first 70% of the data, which is $720 \times 0.7 = 504$ hours, and test

[3]All the code relevant to this paper can be found on https://github.com/HaTT2018/Deep_Gravity.

our model by the rest 30% of the data, which is $720 \times 0.3 = 216$ hours. The results are insightful and detailed in Section 6.

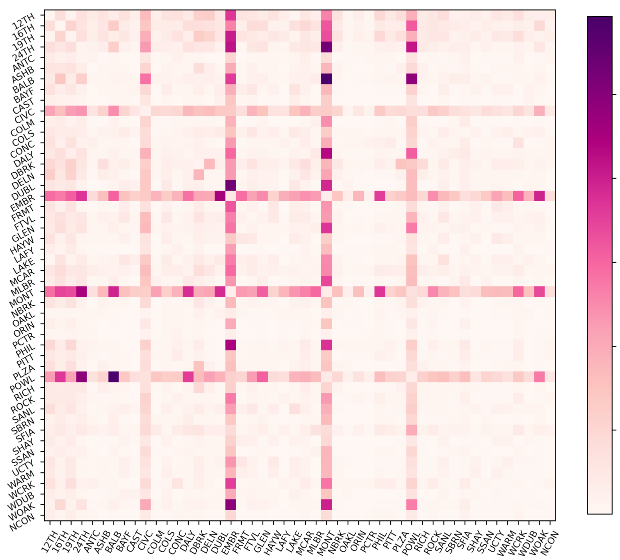


Figure 4: OD matrix

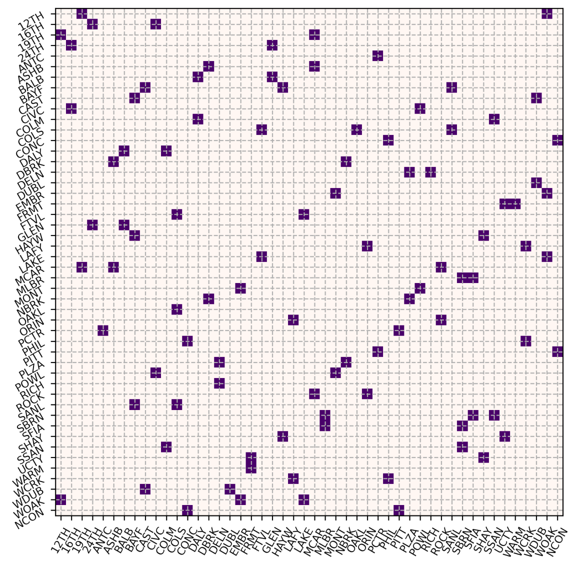


Figure 5: Adjacency matrix

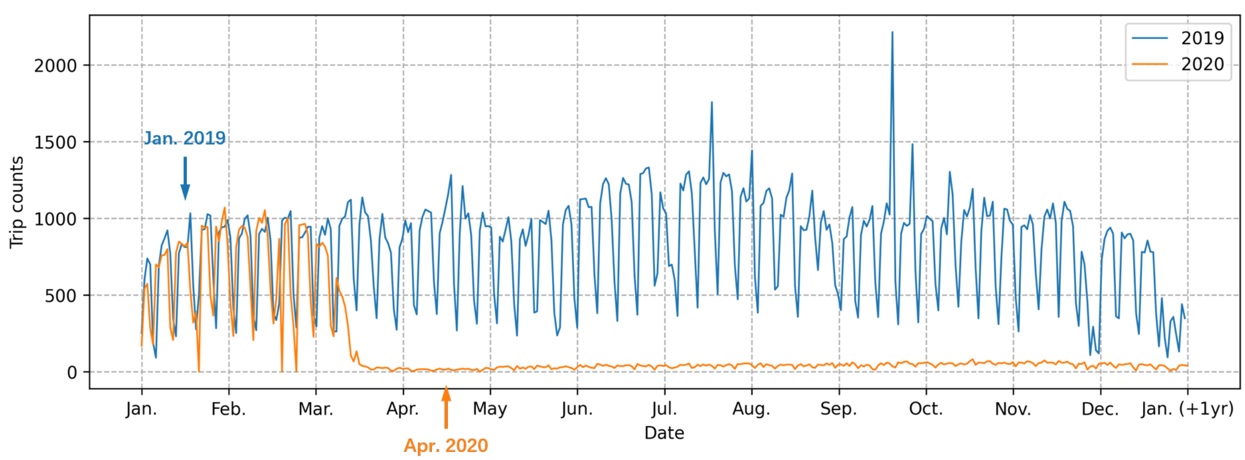


Figure 6: Comparison between daily ridership in 2019 and 2020

5.2 Evaluation metrics

In this paper, we use three metrics to evaluate the performance of ridership prediction models. The first one is the common part of commuters (CPC), or the Sørensen-Dice index (Simini et al., 2021). The index was originally defined as $2|X \cap Y|/(|X| + |Y|)$, i.e., the similarity of two samples is measured by the ratio of 2 times the common part of two samples to the sum of two samples. Denote x_t^j and $\hat{x}_t^j$ for the observed and predicted demands of the jth OD pair in time interval t, respectively, then CPC can be computed as:

$$\mathrm{CPC} = \frac{2\sum_{t,j}\min\left(\hat{x}_t^j, x_t^j\right)}{\sum_{t,j}\hat{x}_t^j + \sum_{t,j}x_t^j} \tag{6}$$

We can find that CPC is contained in $[0, 1]$ with 1 indicating a perfect match between the observed and predicted demands, and 0 indicating the two have no overlap.

Besides, we use two classic deviation measures, namely the normalized root mean squared error (NRMSE) and the normalized mean absolute error (NMAE):

$$\mathrm{NRMSE} = \sqrt{\frac{\sum_{t,j}\left(\hat{x}_t^j - x_t^j\right)^2}{N}} \times \frac{1}{x_{\max} - x_{\min}} \times 100\% \tag{7}$$

$$\mathrm{NMAE} = \frac{1}{N}\sum_{t,j}\left|\hat{x}_t^j - x_t^j\right| \times \frac{1}{x_{\max} - x_{\min}} \times 100\% \tag{8}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum among the observed and predicted demands, respectively, and N represents the number of observed and predicted demand samples.

6 Results and discussions

6.1 Model performance

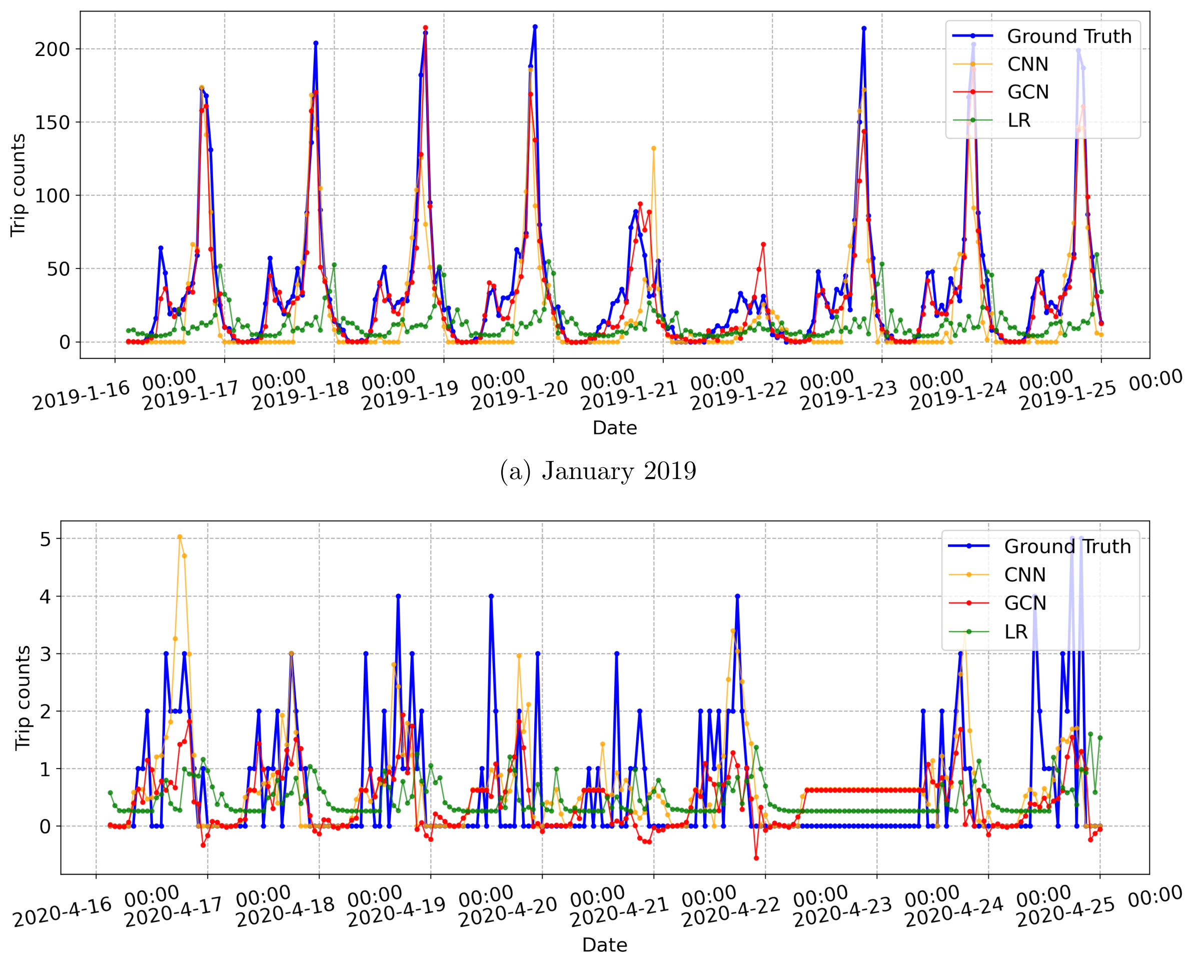
The deviations and CPC values of the three models are listed in Table 3. Taking the demand from EMBR to DBRK as an example, the predictions of trips counts are illustrated in Fig. 7a for the year of 2019 and in Fig. 7b for the year of 2020. In the two figures, the blue line represents the real ridership; the green, orange, and red lines represent the prediction results of the LR, CNN, and GCN models, respectively,

Table 3: Deviations and CPC (sorted by year)

Model	Time	NRMSE	NMAE	CPC
LR		0.0253	0.0072	0.4935
CNN	Jan. 2019	0.0278	0.0127	0.7935
GCN		0.0107	0.0033	0.9220
LR		0.0098	0.0046	0.3338
CNN	Apr. 2020	0.0108	0.0060	0.8290
GCN		0.0090	0.0041	0.8505

GCN can predict the ridership closest to the real ridership and can predict with the highest accuracy. As introduced before, CPC is for comparing the similarity of two datasets. Hence, the higher the CPC value is, the more similar the two datasets are. We can see from Table 3 that the GCN model has the highest CPC value for both 2019 and 2020. Additionally, this value is very close to 1, showing that GCN's predicted ridership is very close to the ground truth. The NRMSE and NMAE values of the GCN model are also the lowest, which indicates that it has the lowest prediction error and therefore the highest accuracy.

We can also analyze the illustrations Figs. 7a and 7b, where the GCN performs steadily and fits the real observations very well. For the LR model, we may conclude from in Table 3 that its deviations are low, yet its predictive performance is not satisfactory according to the illustrations. The green line denoting the LR prediction results keeps in a low ridership level throughout the entire prediction horizon, which demonstrates that it fails to capture the true ridership adequately. Moreover, the CNN model also has disadvantage told from the graphs. There are many odd values whenever it is high ridership or low. This phenomenon indicates that the model is over-estimated, though the validation loss does not increase with respect to training epochs during training the model. The over-estimating problem also shows that a simple convolutional neural network is not suitable for convoluting OD matrix.



(a) January 2019

(b) April 2020

Figure 7: Prediction results of each model before and during the pandemic

Furthermore, the comparison between the prediction results for 2019 and 2020 is also insightful. Table 3 shows that the deviations generally become lower during the pandemic, possibly due to the higher stability of the 2020 data—the ridership remains very low during the pandemic. However, the CPC values do not necessarily increase as the prediction errors decrease, which implies that the ridership patterns are not better captured during the pandemic. This phenomenon is illustrated in Figs. 7a and 7b.

From the previous two paragraphs of discussion, we can summarize that GCN's predictions have low errors, high similarity to the ground truth, and high stability. This can be credit to the usage of the spatial connectivity information of stations, meaning that the model considers not only each origin station itself, but its adjacent stations as well.

Finally, we visualize the observations and the predictions of the GCN model in Fig. 8, where blue, red, and yellow represents low (the lowest 30%), medium, and high (the highest 40%) ridership. We can find from the figure that the model reproduces the observations accurately in general.

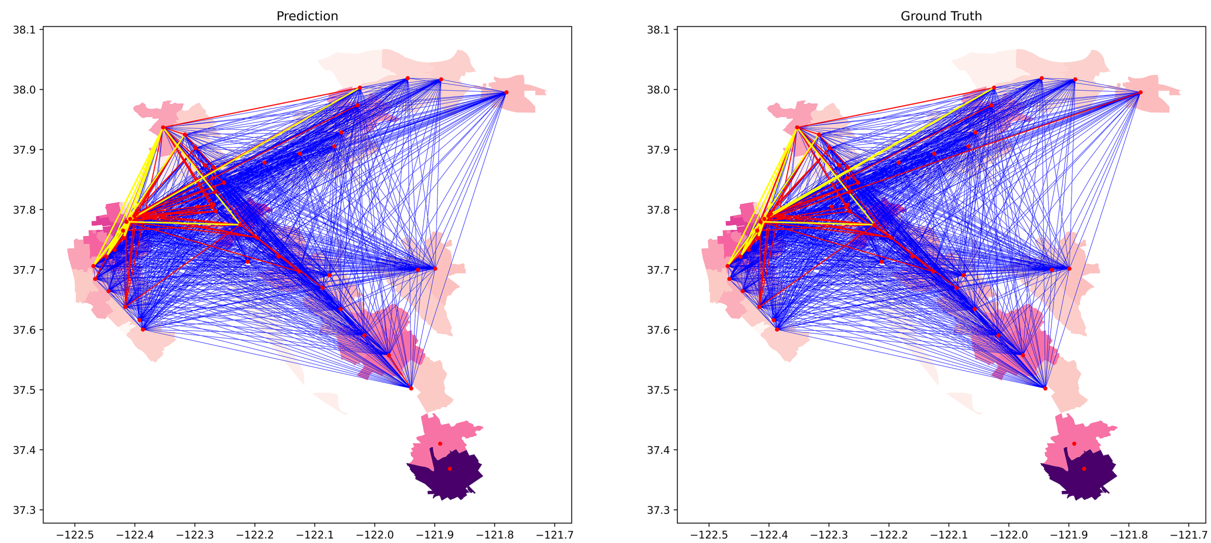


Figure 8: Visualization of the predicted and real OD demands

6.2 Computational efficiency

In addition to the predictive performance, model computational efficiency is also important since there are often trade-offs between the two. Here we analyze the efficiency of each model. Note that we used a different type of hardware for the LR model (i.e., CPU) from that for the other two (i.e., GPU) since they are machine learning based models which require high computational power. The CPU used is 3.20GHz based AMD Ryzen 7 5800H and the GPU used is NVIDIA GeForce RTX 3060 Laptop GPU. The computing time is listed in Table 4.

Table 4 demonstrates that the LR model is the fastest, which is consistent with our intuition because it is simply matrix operations. Nevertheless, as discussed in Section 6.1, linear regression does not provide satisfactory predictions. Thus, we only need to compare the other two more useful models. As shown in Table 4, the computational time of GCN is less than that of CNN by $(126 - 31.8)/126 \times 100\% = 74.8\%$. Therefore, GCN is much more efficient than CNN.

Table 4: Model training time cost

Model	Training time (s)	Hardware used
LR	0.293	CPU
CNN	126	GPU
GCN	31.8	GPU

7 Conclusions and future work

The GCN model outperforms the other two models in both pre-pandemic and pandemic scenarios. It has low deviations and high CPC values, indicating that it can predict the BART ridership very closely to the ground truth. In addition, it does not suffer from the over-estimation problem like the CNN model, which means it can make predictions more steadily. Finally, the training process of GCN model is faster than that of CNN. Therefore, GCN is more cost-efficient from the aspect of computational efficiency compared with the CNN model. By contrast, the LR model does not have satisfactory prediction results, although it is simpler to implement than the two machine learning based approaches.

Our study shows that GCN has its outstanding superiority for predicting BART ridership. As the traffic demands are recovering from the pandemic, developing a GCN model may be a wise choice for BART to schedule the trains scientifically. However, it remains questionable how this model can be generalized elsewhere and broadly applied in the industry. Therefore, one of our future directions is to investigate on the transferability of the model to different rapid transit systems (e.g., the New York City Subway). Additionally, we can also study how the demand prediction model can help decrease the operation costs of transit authorities and benefit the society.

References

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). Spectral Networks and Locally Connected Networks on Graphs. *ArXiv:1312.6203 [Cs]*. https://arxiv.org/abs/1312.6203

Chen, P., Fu, X., & Wang, X. (2021). A graph convolutional stacked bidirectional unidirectional-LSTM neural network for metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2021.3065404

Clark, S. (2003). Traffic Prediction Using Multivariate Nonparametric Regression. *Journal of Transportation Engineering*, *129*(2), 161–168. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:2(161)

Du, B., Hu, X., Sun, L., Liu, J., Qiao, Y., & Lv, W. (2021). Traffic demand prediction based on dynamic transition convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, *22*(2), 1237–1247. https://doi.org/10.1109/TITS.2020.2966498

Hamed, M. M., Al-Masaeid, H. R., & Said, Z. M. B. (1995). Short-Term Prediction of Traffic Volume in Urban Arterials. *Journal of Transportation Engineering*, *121*(3), 249–254. https://doi.org/10.1061/(ASCE)0733-947X(1995)121:3(249)

Jin, G., Cui, Y., Zeng, L., Tang, H., Feng, Y., & Huang, J. (2020). Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. *Transportation Research Part C: Emerging Technologies*, *117*, 102665. https://doi.org/10.1016/j.trc.2020.102665

Ke, J., Zheng, H., Yang, H., & Chen, X. (Michael). (2017). 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. https://doi.org/10.1016/j.trc.2017.10.016

Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. *ArXiv:1609.02907 [Cs, Stat]*. http://arxiv.org/abs/1609.02907

Lee, S., & Fambro, D. B. (1999). Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting. *Transportation Research Record: Journal of the Transportation Research Board*, *1678*(1), 179–188. https://doi.org/10.3141/1678-22

Liu, L., Chen, J., Wu, H., Zhen, J., Li, G., & Lin, L. (2020). Physical-Virtual Collaboration Modeling for Intra- and Inter-Station Metro Ridership Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 1–15. https://doi.org/10.1109/TITS.2020.3036057

Lu, B., Gan, X., Jin, H., Fu, L., & Zhang, H. (2020). Spatiotemporal Adaptive Gated Graph Convolution Network for Urban Traffic Flow Forecasting. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 1025–1034. https://doi.org/10.1145/3340531.3411894

Ma, X., Zhang, J., Du, B., Ding, C., & Sun, L. (2019). Parallel Architecture of Convolutional Bi-Directional LSTM Neural Networks for Network-Wide Metro Ridership Prediction. *IEEE Transactions on Intelligent Transportation Systems*, *20*(6), 2278–2288. https://doi.org/10.1109/TITS.2018.2867042

Okutani, I., & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1), 1–11. https://doi.org/10.1016/0191-2615(84)90002-X

Simini, F., Barlacchi, G., Luca, M., & Pappalardo, L. (2021). A Deep Gravity model for mobility flows generation. *Nature communications*, *12*(1), 1–13. https://doi.org/10.1038/s41467-021-26752-4

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for Deep Spatial-Temporal Graph Modeling. *ArXiv:1906.00121 [Cs, Stat]*. http://arxiv.org/abs/1906.00121

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 3634–3640. https://doi.org/10.24963/ijcai.2018/505

Zhang, S., Zhou, L., Chen, X. (Michael), Zhang, L., Li, L., & Li, M. (2020). Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition. *Computer-Aided Civil and Infrastructure Engineering*, *35*(10), 1132–1147. https://doi.org/10.1111/mice.12575

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., & Li, H. (2020). T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(9), 3848–3858. https://doi.org/10.1109/TITS.2019.2935152