CitiBike Demand Prediction by Mining Spatial-temporal Pattern

Yu Tang, Qian Xie

December 19, 2019

Executive Summary

Our project focuses on predicting station-level demands in the next hour for CitiBike in New York City by capturing the spatial-temporal pattern. We choose Graph Convolutional Network (GCN) as the methodology since this kind of model excels in dealing with graph-structured data and better recognizing correlations among the network. Unlike the previous work which only utilizes the historical data, we also take weather factors and day of the week information into account through a data fusion process. Furthermore, we propose four types spatial dependency matrices for the design of adjacency matrix which is the key input fed into GCN. We also discuss the selection of spatial-temporal parameters. The results show that the demand correlation matrix is the best candidate among the four matrices. Besides, we compare the proposed methodology with two classical prediction models: linear regression and multi-layer perceptron. The comparison shows that our model performs better than the benchmarks in terms of mean squared error.

1 Introduction

Bikes are still popular over the world, thanks to their contributions to fewer emissions, congestion mitigation and so on. Such popularity has been witnessed by bike-sharing systems [1]. From 2013, Citi Bike becomes one of the largest bike-sharing systems with more than 700 docking stations in New York City and Jersey City. It is reported that Citi Bike supports tens of thousands of rides every day [2].

The huge biking demands are challenging the management of bike-sharing systems. One problem is how to deploy bikes efficiently and economically. For the bike-sharing system with dockers, there can be a supply shortage at major stations in rush hours. Meanwhile, relocating bikes requires a large consumption of manpower and expense. This problem might be solved by predicting biking demands since the accurate prediction can guide bike rebalancing.

Currently, most bike-sharing systems publish their operational data online. Given these historical data, we propose to building a real-time prediction model for biking demands. Obviously, the demands at each station have both spatial and temporal dependencies, and they are easily influenced by weather, weekday and off-peak/peak hours. Thus, we plan to use graph convolution neural network (GCN) to tackle this prediction problem. The advantages of GCN are twofold. First, it models bike stations as nodes in a graph which means it can capture the spatial and temporal connection among bike stations in terms of network [3]. Second, neural network models are good at fusing heterogeneous data and mining potential relationships.

The rest is organized as follows. Section 2 reviews the literature on bike-sharing demand prediction models. Then our prediction problem and methodology are identified in section 3 and 4, respectively. Section 5 describes the process of data collection. The data analysis is presented in section 6. The following section illustrates the details of our model. Finally, comparisons with benchmarks are summarized in section 8.

2 Literature Review

The station-level demands in bike-sharing system are characterized by spatial-temporal correlation. Thus how to exploit this character is the key step in demand prediction. Currently, three classes of methodologies

have been proposed to deal with the spatial-temporal features in bike-sharing system. They are 1) cluster-based prediction, 2) spatial regression model and 3) deep neural network.

Cluster-based prediction methods are based on the idea that stations may have some correlation in geographical locations and temporal demands. Froehlich et al. applied clustering techniques to identify shared behaviors across stations and show how these behaviors relate to geographical location, neighborhoods, and time of day [4]. Li et al. proposed a bipartite clustering algorithm to cluster bike stations according to their geographical locations and historical transition patterns, then predicted the number of rentals by GBRT and inferred the number of returns for each station based on their estimated proportion of rentals [5]. Chen et al. modeled the relationship among bike stations with a weighted correlation network and dynamically grouped neighboring stations with similar bike usage patterns into clusters. Their clustering considers not only time of the day, but also weather condition and social events [6]. Bao et al used K-means algorithm to cluster stations and then employed Latent Dirichlet Allocation to mine the travel patterns [7].

Spatial regression models are commonly used to mine the relationships between neighborhoods. Several studies adopted this methodology to analyze the spatial and temporal dependencies in bike-sharing system. Faghih-Imani et al. investigated the interactions between departure and arrival rates with spatial lag model [8]. Singhvi et al. provide pairwise predictions of the demand for trips between each origin-destination pair by running log-log regression models with covariates including taxi usage, weather and spatial variables [9]. Rudloff et al. employed count models with considering whether the three closest stations are empty or full [10].

However, the factors that affect bike demands are complex. To capture the underlying relations between demands and variables and make more accurate predictions, researchers started to adopt deep learning approaches, especially deep neural networks. Among them, Convolutional Neural Network (CNN) performs well in recognizing spatial and temporal patterns. However, CNN is defined and can be applied directly to data with regular grids such as images. For station-level bike demands prediction, since the study area may have an irregular shape, some data preprocessing work have to be done. Zhang et al. split the study area into grids with a pre-defined size. The size can not be too large or too small, which is a drawback of CNN.

Another state-of-art deep neural network is Graph Convolutional Neural Network (GCN). GCN is a good candidate for our task, since it is powerful in dealing with graph-structured data and bike-sharing networks can be represented by nodes and arcs. There are already some studies in transportation fields using GCN to make predictions, Xiong et al. proposed a novel O-D prediction framework combining GCN and Kalman filter to identify spatial and temporal patterns simultaneously [11]. Only historical data are used. Lin et al. used GCN with data-driven graph filter to learn hidden heterogeneous pairwise correlations between stations [12]. Both research only consider historical data.

In our project, GCN will be applied to make station-level predictions on both renting number and returning number and then derive station-level bike demands. Besides endogenous variables (historical data), we will also incorporate exogenous variables such as weather and weekday.

3 Problem Description

We aim to predict station-level demands of each bike station in Lower Manhattan. Concretely, we forecast the number of rentals at bike stations in the next hour. This prediction can guide the relocating strategy given the current spatial distribution of bikes.

We use historical data from 2019/01/01 to 2019/10/31 to train the prediction model. The problem is illustrated in Fig.1. The training data includes historical Citi Bike records, weather data and time factors. Historical Citi Bike records include the hourly number of renting and returning of each bike station. The weather data include temperature, precipitation, humidity and so on. Time factors include the corresponding weekday and hour. The prediction model outputs the number of rental in the next hour for each station.

4 Proposed Methodology

Our methodology is illustrated in Fig.2. The model first takes the previous rental and return data as inputs, which are in the form of graph. Then it achieves data fusion with weather and day of the week information through fulled-connected network. Finally, it outputs the demand of each bike station in the next hour.

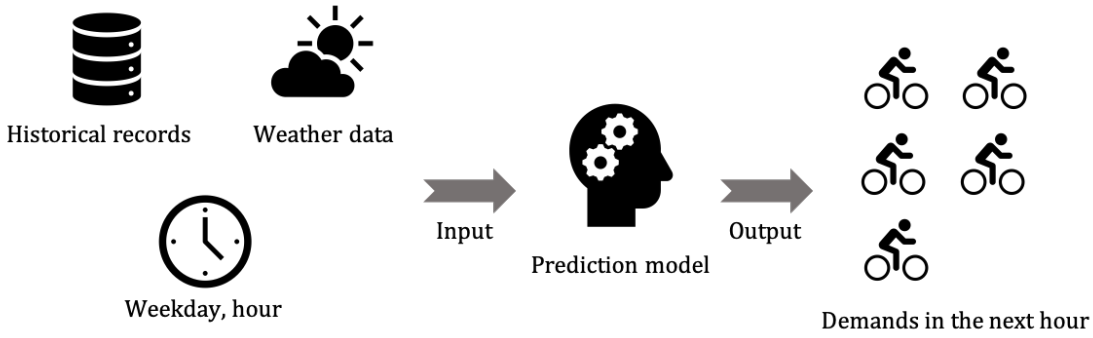

Figure 1: Prediction problem

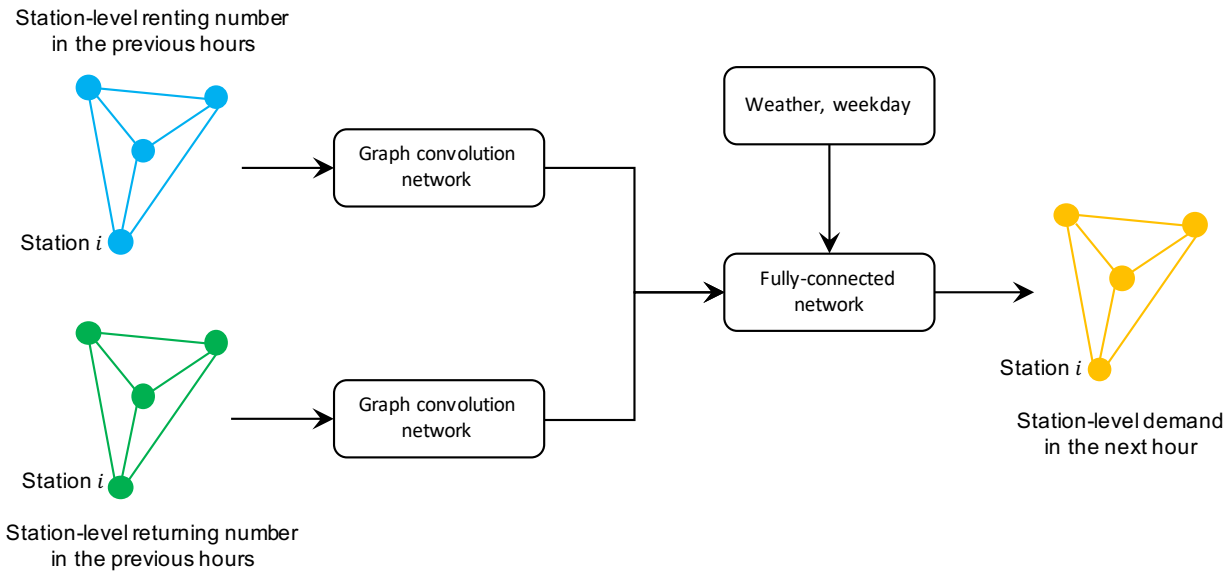

Figure 2: Methodology framework

First, we represent Citibike data with graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ where node set $\mathcal{N}$ denotes bike stations and link set $\mathcal{E}$ denotes connectivity between bike stations. The key point lies in how to construct the adjacency matrix $A \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$ that measures the dependencies between nodes. We will discuss the dependency representation in terms of geographic distance, daily demand, trip duration and demand correlation.

Then we pass the processed Citibike data into graph convolution networks (GCN). Let $H^{(0)} = [X_1, \cdots, X_{|\mathcal{N}|}]^\intercal \in \mathbb{R}^{|\mathcal{N}|\times N_f}$ be the input of GCN where X_n denotes the feature of bike station n and N_f denotes the feature size. Herein N_f equals the hour number used for prediction. The propagation between GCN layers is given by

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

where $H^{(l)}$ is the input of layer l/the output of layer $l-1$, D is the diagonal matrix of A, $W^{(l)}$ is the weight between layer l and $l+1$, $\sigma(\cdot)$ is the activation function.

We fulfill the data fusion with weather and weekday data through fully-connected network. The output of GCN is flattened and concatenated with weather data, which is taken as the inputs of fully-connected network (FCN). The propagation in FCN is given by

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

where $H^{(l)}$ is the input of layer l/the output of layer $l-1$, $W^{(l)}$ is the weight between layer l and $l+1$, $\sigma(\cdot)$ is the activation function.

Mean squared error is adopted as the loss function $\mathcal{L}$ (see Eq.3)

$$\mathcal{L} = \frac{1}{|\mathcal{N}||\mathcal{T}|} \sum_{n\in\mathcal{N}, t\in\mathcal{T}} (y_{nt} - \hat{y}_{nt})^2 \tag{3}$$

where y_{nt} (resp. $\hat{y}_{nt}$) is the true (resp. predicted) hourly demands of station n at time t, and $\mathcal{T}$ is the set of time points. The model will be optimized by PyTorch.

5 Data Collection

5.1 Citi Bike Data

Since we aim to predict the biking demands in Lower Manhattan, we first figure out 128 bike stations in the scope, which are presented in Fig.3.

The raw Citi Bike data is trip-based, including start time, end time, start station, end station, bike id and so on. We aggregate the trip data to obtain hourly rental and return numbers for all remaining stations. After the preprocessing, we have the station-level rental data as shown in Tab.1. So are the return data.

Table 1: Hourly station-level rental data

	Station ID			
	79	82	$\cdots$	3812
2019/01/01 00:00	1	0	$\cdots$	0
2019/01/01 01:00	0	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
2019/10/31 23:00	0	1	$\cdots$	10

5.2 Weather Data

We use the API provided by https://www.worldweatheronline.com/ to access weather data. The demo code is given below where the parameter 'tp' denotes time interval (unit: hour). Herein we collect weather records every hour from 2019/01/01 to 2019/10/31.

Figure 3: Bike stations in Lower Manhattan

```
import requests
# Set query parameters
params = {'q': 'new york', 'tp': 1, 'key': myKey, 'date': '2019-{}-{}'.format(month, day)}
# Obtain weather data in format of xml
weatherData = requests.get('https://api.worldweatheronline.com/premium/v1/past-weather.ashx'
    , params)
```

Listing 1: Collect weather data

The historical weather data are stored in the format of xml. We extract 1) temperature, 2) precipitation, 3) humidity, 4) heat index temperature, 5) wind speed and 6) weather condition description from the raw data, as shown in Tab.2. The first five factors are numerical and the last is categorized.

Table 2: Hourly historical weather data

	temp	precip	humidity	heat index	wind speed	weather condition
2019/01/01 00:00	0	9.3	89	2	10	heavy rain
2019/01/01 01:00	3	4.6	91	5	10	heavy rain
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2019/10/31 23:00	14	0.8	77	16	26	moderate rain

6 Data Analysis

6.1 Data Quality

We first investigate the boxplot of hourly rental and return. The circles in Fig.4 denote the outliers. Although there are lots of outliers for both rental and return, it is noticed that the distributions of rental and return are similar. So the outliers might reliable and they indicate some stations have large demands or are influenced by some special events.

We also observe that all stations have complete records from 2019/01/01 to 2019/10/31. As shown in Figure 5, some stations are newly opened (e.g. Station 3812) while some are temporarily "closed" (e.g.

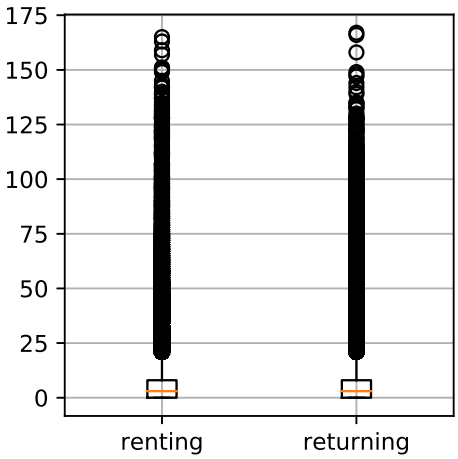

Figure 4: Distribution of hourly number of rental and return

Station 368). We delete those stations with more than 30% missing data and create a new binary feature "isMissing".

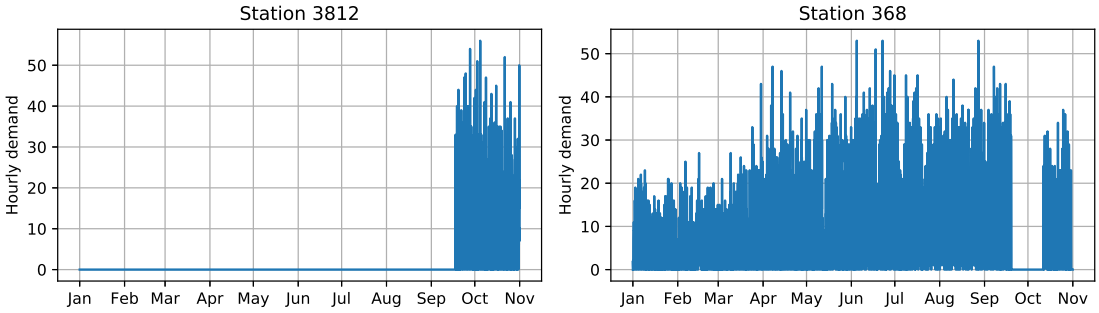

Figure 5: The number of historical records at Station 3812 and Station 368

Next, we scrutinize the weather data by focusing on the historical temperature. Figure 6 presents the hourly temperature and heat index from 2019/01/01 to 2019/10/31. It can be found some temperature in September equals zero but the corresponding heat index is still high. These weather data might be abnormal and should be removed from the training data.

6.2 Weather impact

To figure out the impact of weather on average rental number and return number, we draw a series of scatter plots for each weather statistics. The pattern of the data shows the positive relationship between temperature/heatindex/windspeed and rental/return number as well as the negative relationship between humidity/percipitation and rental/return number. See Figure 7.

6.3 Temporal Feature

We also study the relationship between temporal features and rental/return number. For each station, we groupby the data by hour of the day and calculate the mean for all stations, then compute their mean to obtain the average rental number and average return number. The plot of the relationship between average rental/return number and hour of the day shows bimodality. See Figure 8.

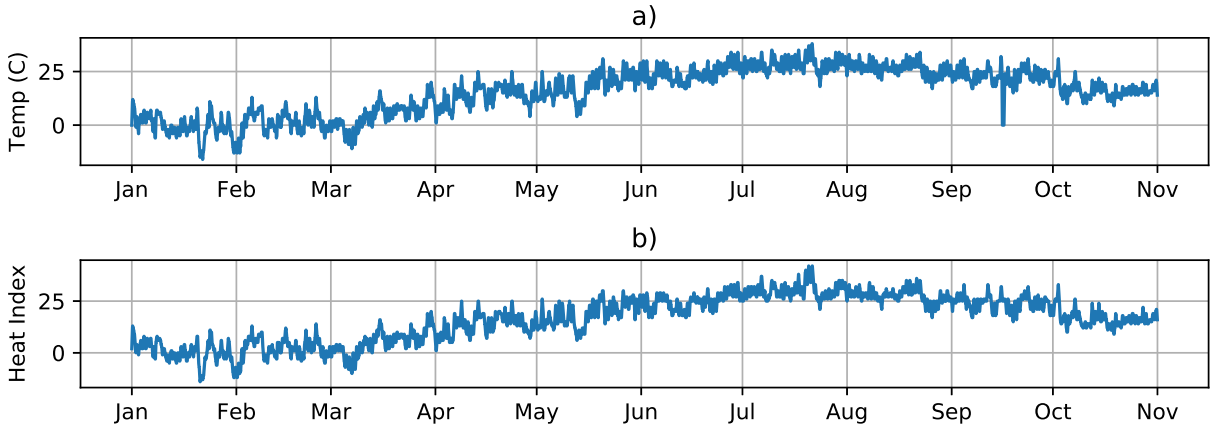

Figure 6: Weather data

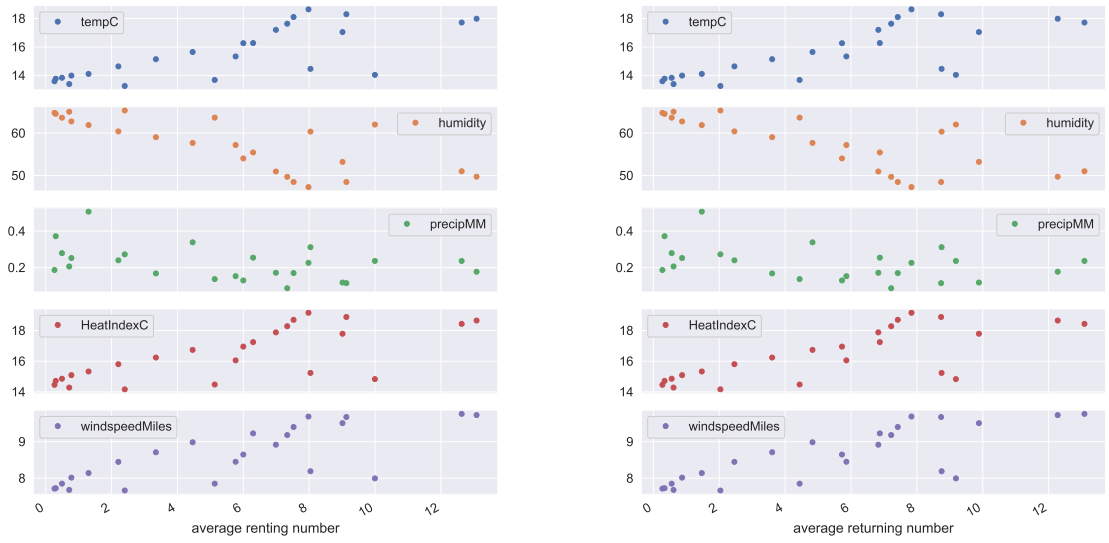

Figure 7: Relationship between average rental/return number and weather statistics

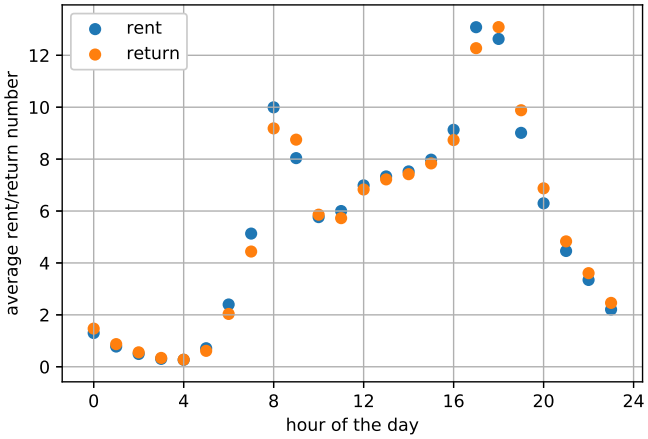

Figure 8: Average rental number and return number by hour of the day

6.4 Spatial Dependency

The spatial dependency is also an aspect that we want to focus, so we compute the correlation matrix of time series rental numbers for the 128 stations and plot the heatmap based on the magnitude of correlation. Lower the correlation, deeper the color. This correlation plot shows certain spatial dependency among citibike stations since the colors are not random. See Figure 9.

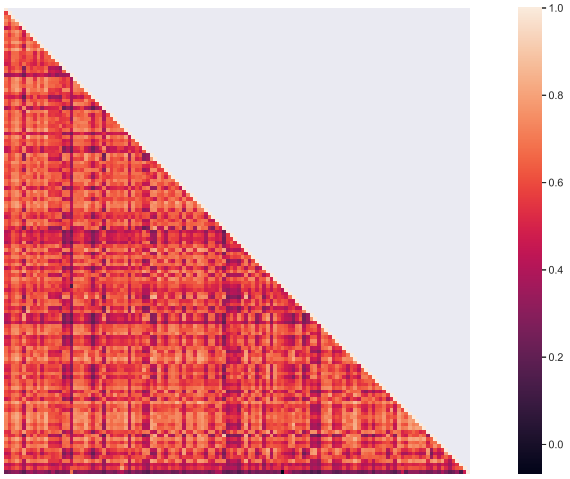

Figure 9: Correlation between station-level rental number

7 Model Analysis

7.1 Adjacency Matrices

As mentioned in Section 4, the adjacency matrix A needs to be defined before we run GCN. Hence we use four kinds of matrices to construct adjacency matrices and compare their performances on training set and test set.

7.1.1 Geometric distance matrix (GD)

One way to represent the spatial correlation between stations is simply through the geometric distance, i.e. the shortest spherical distance between the geographical locations (determined by the latitude and longitude) of two stations. For any two stations, if they are close to each other in terms of geometric distance, they are connected in the GCN. Now the adjacency matrix A is built such that $A_{ij} = 1$ if $dist(i, j) \geq \theta_{GD}$ and 0 otherwise.

7.1.2 Daily demand matrix (DD)

Each entry of DD is obtained from aggregating average daily OD demand, which follows a long-tail distribution. For adjacency matrix A, we also use a threshold θ_{DD} to determine the connection. If the average daily demand between station i and station j is above θ_{DD}, then A_{ij} is set to 1; otherwise, A_{ij} should be 0.

7.1.3 Trip duration matrix (TD)

The average trip duration is calculated as total trip duration / total OD demand. Similarly, we connect those two stations based on the average trip duration between them. If lower than the threshold θ_{TD}, then $A_{ij} = 1$; otherwise $A_{ij} = 0$.

7.1.4 Demand correlation matrix (DC)

We also consider the correlation coefficient between hourly demand series, which has a range of [0, 1] to capture the relationship of two stations. The demand correlation matrix has already been computed in subsection 6.4. For those high correlation coefficients $DC_{ij} \geq \theta_{DC}$, the entry A_{ij} is 1 accordingly.

7.1.5 Performance Comparison

We first construct GD, DD, TD and DC matrices from the raw data. The distribution of the elements in the matrices are presented in Figure 10.

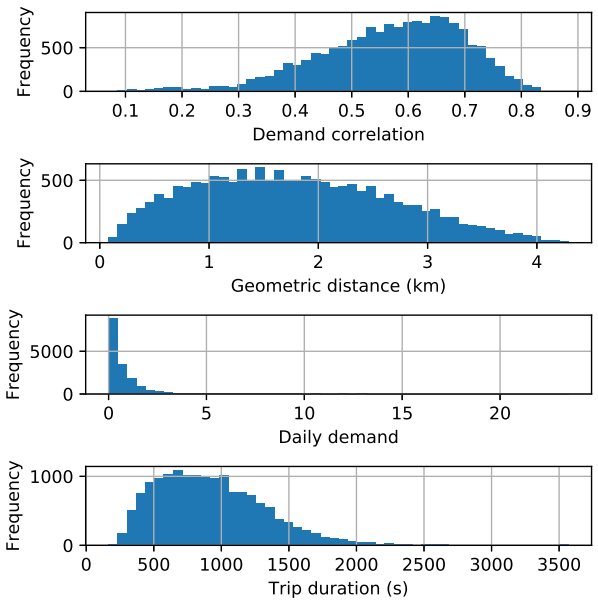

Figure 10: Distribution of the elements in the four matrices.

We use the four matrices to train GCN separately. The training loss is presented in Fig 11.

After the training, we evaluate the model performance in the testing dataset. The selected metric is mean squared error (MSE). Table 3 shows that DC outperforms other matrices.

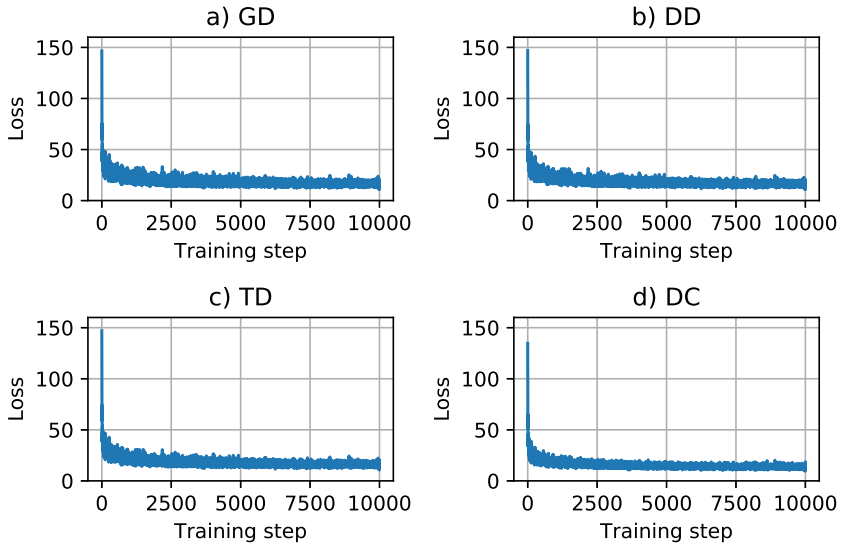

Figure 11: Training loss

Table 3: Comparison in terms of MSE

	GD	DD	TD	DC
Train	13.0	12.3	13.1	11.3
Test	18.4	18.0	18.3	17.9

7.2 Impacts of spatial-temporal features

We choose the GCN based on demand correlation matrix (DC) to investigate the impacts of spatial-temporal features.

7.2.1 Impacts of spatial dependencies

As indicated in section 7.1, the proposed GCN requires a threshold to generate adjacency matrices. Obviously, the threshold value influences the quantification of spatial dependencies. For the demand correlation matrix, the element values vary from 0 to 1. If θ_{DC} is high, fewer stations are considered as "neighbours". It encourages GCN to consider each stations more independently. But low θ_{DC} results in more "neighbours". It might induce to GCN to mine the spatial dependencies from the stations that are weakly related. Table 4 shows that θ_{DC} should be neither too low nor too high.

Table 4: Impacts of threshold θ_{DC}

θ	0.1	0.5	0.8	1.0
Train	13.9	13.3	11.3	14.0
Test	18.3	18.2	17.9	19.1

7.2.2 Impacts of temporal features

We also tried different number of previous hours (denoted as T). Table 5 shows that using 2 previous hours for prediction can have a better performance.

Table 5: Impacts of temporal features

T	1	2	3
Train	12.6	11.3	12.7
Test	18.9	17.9	18.2

8 Model Comparison

For comparison, we choose two classic benchmarks. One is Linear Regression (LR) and the other is Multi-layer perceptron (MLP). Linear regression is a linear approach to explaining the relationship between dependent variable (next hour demand) and independent variables (historical demands and weather factors). Multi-layer perceptron, or fully-connected network, is the most typical neural network. Both LR and MLP take the historical data of all stations and other factors as the model input and then predict the future demands of each station. The selected MLP has one hidden layer with 200 neurons.

Tab 6 presents the comparison in terms of the number of model parameters and MSE, and Fig 12 illustrates the prediction quality. Usually, the model performance increases in the number of model parameters. So it is not surprising to find that GCN outperforms LR. However, it is interesting to find that GCN with fewer parameters is still better than MLP. This finding demonstrates that GCN is more efficient at mining spatial-temporal patterns.

Table 6: Model comparison

	LR	MLP	GCN
# Parameters	0.66×10^5	1.37×10^5	1.33×10^5
Train	11.1	12.0	11.3
Test	22.6	19.3	17.9

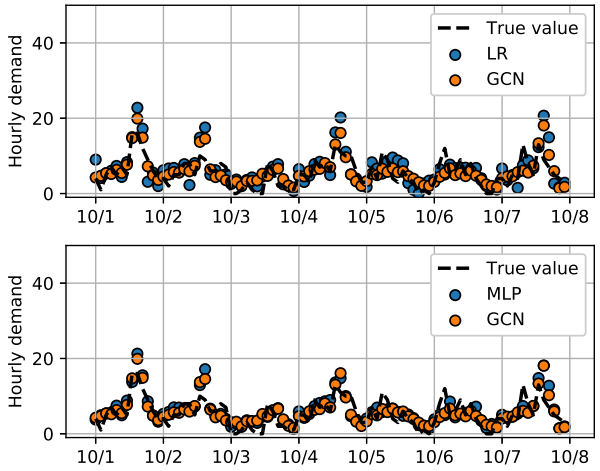

Figure 12: Prediction at station 79

9 Conclusion

In this project, we design GCN for the prediction of CitiBike demands in Lower Manhattan, New York City. Our project can benefit bike rebalancing and management. The prediction problem is described as

forecasting the demands at each station in the next hour given the historical records and weather data. After collecting and preprocessing the Citi Bike Trip Histories and weather data, we discover that there are certain spatial-temporal patterns underlying the hourly station-level rental/return data and weather also has an impact on the demand. We propose a GCN model with a data fusion process. After testing four types of matrices: GD, DD, TD and DC for the design of adjacency matrices which plays an important role in GCN, we found that DC can best characterize the spatial dependencies among stations and 0.8 is a better threshold to determine the connection of stations in the network. In terms of the temporal features, we also study the suitable number of previous hours for prediction. Experiments show that using 2 previous hours is a good alternative. After tuning the parameters, we compare our GCN model to two classic benchmarks: Linear Regression (LR) and Multilayer Perceptron (MLP) from both qualitative and quantitative perspective. All comparisons mentioned above are based on mean squared error. The results prove that our model outperforms LR and MLP.

For future direction, it may be helpful to consider social events, land use and constructions around bike stations. Besides, our project is limited to Lower Manhattan and there would be more interesting results when we study the whole New York City. For example, there are long trips across boroughs and short trips within neighbourhoods. Trips in different boroughs can have different patterns. The trip types (commute or tourism) can also play an important role.

References

[1] E. Fishman, S. Washington, and N. Haworth, "Bike share: a synthesis of the literature," *Transport reviews*, vol. 33, no. 2, pp. 148–165, 2013.

[2] Y. Chen, Z. Liu, and D. Huang, "Unlock a bike, unlock new york: A study of the new york citi bike system," in *International Conference on Intelligent Interactive Multimedia Systems and Services*. Springer, 2018, pp. 356–366.

[3] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[4] J. E. Froehlich, J. Neumann, and N. Oliver, "Sensing and predicting the pulse of the city through shared bicycling," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[5] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 33.

[6] L. Chen, D. Zhang, L. Wang, D. Yang, X. Ma, S. Li, Z. Wu, G. Pan, T.-M.-T. Nguyen, and J. Jakubowicz, "Dynamic cluster-based over-demand prediction in bike sharing systems," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 841–852.

[7] J. Bao, C. Xu, P. Liu, and W. Wang, "Exploring bikesharing travel patterns and trip purposes using smart card data and online point of interests," *Networks and Spatial Economics*, vol. 17, no. 4, pp. 1231–1253, 2017.

[8] A. Faghih-Imani and N. Eluru, "Incorporating the impact of spatio-temporal interactions on bicycle sharing system demand: A case study of new york citibike system," *Journal of Transport Geography*, vol. 54, pp. 218 – 227, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0966692316303143

[9] D. Singhvi, S. Singhvi, P. I. Frazier, S. G. Henderson, E. O'Mahony, D. B. Shmoys, and D. B. Woodard, "Predicting bike usage for new york city's bike sharing system," in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[10] C. Rudloff and B. Lackner, "Modeling demand for bikesharing systems: Neighboring stations as source for demand and reason for structural breaks," *Transportation Research Record*, vol. 2430, no. 1, pp. 1–11, 2014.

[11] X. Xiong, K. Ozbay, L. Jin, and C. Feng, "Dynamic origin-destination matrix prediction with line graph neural networks and kalman filter," *arXiv preprint arXiv:1905.00406*, 2019.

[12] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.