Predicting Bike Sharing Usage As The Network Expands

Alexander Guo
Massachusetts Institute of Technology
akguo@mit.edu

Timothy Tay
Massachusetts Institute of Technology
tayt@mit.edu

1. Introduction

The demand and supply of urban mobility services are highly interdependent. However, the relationship between demand and supply is also highly nonlinear, which makes it difficult to study analytically. For example, in the case of bike sharing, the opening of a new station not only has the effect of drawing demand away from nearby surrounding stations, it also attracts new members to join the bike sharing scheme, thereby most likely increasing the overall demand for bike sharing. Depending on where this station is opened (i.e. if it is already in an area with many stations, or if it is badly needed), these different effects will impact demand for this new station and surrounding stations differently.

If it is possible to accurately predict the effects of opening, or even closing, bike sharing stations on the demand for bike sharing at a system level (or at least, at the new station and surrounding stations), we would have a very useful planning tool for bike sharing operations. This idea can also be applied to other forms of shared mobility such as car sharing. It may even be useful to study the effects on demand of opening and closing bus or subway stations.

2. Related Work

The bike sharing demand prediction problem is a problem for which many prediction models have been proposed. Some of the proposed models attempt to predict demand at the cluster level [2, 9], using more classical methods involving identifying clusters of stations, before predicting demand for the clusters. For instance, Chen *et al*. [2] use a weighted correlation network to model and dynamically group neighboring stations with similar usage patterns into clusters. The over-demand probability at each cluster is then predicted using Monte Carlo simulations. Li *et al*. [9] make use of a bipartite clustering algorithm to cluster bike stations into groups, before applying a Gradient Boosting Regression Tree to predict the bike sharing demand across the whole city. The city-level demand is then split across clusters based on a multi-similarity-based inference model.

Recently, more studies have been tackling the more challenging problem of predicting bike sharing demand at the station level [11, 16, 18]. Liu *et al*. [11] developed a Meteorology Similarity Weighted K-Nearest-Neighbor regressor to predict station-level demand. Yang *et al*. [16] used a random forest regressor to predict the number of bicycle checkouts, but without taking into account any spatial or temperoal correlations. Zhang *et al*. [18] proposed the *ST-ResNet*, which is comprised of convolutional neural networks (CNN) layers to model the spatial correlations and a residual neural network framework to model the temporal components, to predict station-level demand.

While CNN can be used to model spatial correlations, it requires the data to reside in a regular domain. To overcome this limitation, the graph neural network (GNN) was proposed [13], along with graph convolutional networks (GCN) which generalizes convolutions to the graph domain. One approach is to define the convolution operation in the Fourier domain by working with the spectral representation of the graph [1, 14]. However, the convolution operation for graphs is computationally intensive. To overcome this, Defferrard *et al*. [5] proposed a fast localized spectral filtering approach called *ChebNet*. Kipf *et al*. [8] further simplified the convolution operation by using only the first-order approximations. A review of other types of graph convolutional networks can be found in [19].

Following the early successes of GCN, many studies have applied GCN to transportation problems, including ride-hailing demand forecasting [6], traffic flow predictions [17] and origin-destination matrix prediction [15]. GCN has also been used for bike sharing demand prediction [7, 10], albeit while working with a static graph, where no new stations are added to the network being studied.

3. Method

3.1. Problem Formulation

The objective of this study is to predict the changes in bike sharing spatiotemporally, as a result of opening new stations. Previous studies have proposed the use of CNNs to model the spatial correlations when predicting bike sharing usage (see *e.g*. Zhang *et al*. [18]). This involves dividing the

area of study into grids. However, in the case of bike sharing, grids are not an ideal representation of the problem. If the grid size is too big, multiple bike sharing stations would end up in the same grid square. However, if the grid size is reduced to ensure at most one station per grid square, the dimension of the grid would be immensely large, making the problem computationally very expensive to tackle.

To overcome these issues, we use a GCN [8] to model spatial correlations instead. The bike sharing stations is represented as a weighted, undirected graph $G^{(t)}(V, E^{(t)})$ at time step t, where V is the set of nodes representing the bike sharing stations, and $E^{(t)}$ is the set of edges at time step t. Note that V remains constant through time, and includes both existing and new stations which opened in 2016.

For each node $v \in V$, we define a feature vector $x_v^{(t)} \in \mathbb{R}^d$ at each time step t, where d is the number of features used. This is then used to define the feature matrix $X^{(t)} \in \mathbb{R}^{|V| \times d}$ at every time step t. The features to be used at every node for prediction include:

- Latitude and longitude of the station
- Hourly number of trips starting from the station
- Hourly net usage at the station
- Hour of the day
- Day of the week
- Month
- Maximum temperature of the day
- Minimum temperature of the day
- Total amount of rain during the day
- Total amount of snow during the day

The weight of an edge $\beta^{(t)}(v_i, v_j)$ at time step t can be computed based on a pre-defined measure of correlation between the nodes v_i and v_j. The adjacency matrix of the graph $G^{(t)}$ is then given by:

$$A_{ij}^{(t)} = \beta^{(t)}(v_i, v_j). \tag{1}$$

Based on the results of Lin *et al*. [10], a GCN using a graph built based on demand correlation is highly effective at learning to predict station-level hourly demand. However, when considering new stations, it is difficult to get an estimate of how demand at the new station would correlate with demand at existing stations. Hence, in this study, we consider graphs constructed based on the spatial distance between stations. More specifically, an edge between two stations exists if the distance between the two stations is less than a pre-specified threshold κ. If the edge exists, it is assigned a weight computed based on the inverse distance between stations (*i.e*. greater weight is assigned to edge between station pairs that are closer to each other):

$$\beta^{(t)}(v_i, v_j) = \begin{cases} \frac{1}{\text{dist}(v_i, v_j)} & \text{if dist}(v_i, v_j) < \kappa \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Furthermore, the edge weights are a function of time, in that all edges leading to/from stations which have not been opened have a weight of zero. A non-zero weight is only assigned from the time step in which the station is opened.

Having computed the adjacency matrix $A^{(t)}$, the inputs of the GCN are propagated to the output with the rule:

$$H^l = \sigma\left(\left(D^{(t)}\right)^{-\frac{1}{2}}\left(A^{(t)} + I_{|V|}\right)\left(D^{(t)}\right)^{-\frac{1}{2}} H^{l-1}W^l\right) \tag{3}$$

where $H^l \in \mathbb{R}^{|V| \times d^l}$ is the matrix of activations in the l^{th} layer, containing d^l hidden units, $D^{(t)}$ is the modified degree matrix (*i.e*. $D_{ii}^{(t)} = \sum_j \left(A_{ij}^{(t)} + \delta_{ij}\right)$), $I_{|V|}$ is the identity matrix of dimension $|V|$, and W^l is the learnable weight matrix. For more details about the derivation of Eq. (3), we refer the reader to [8] and Section 3.1 of [10].

The prediction problem can then be defined as such:

$$\left[Y^{(t)}, \ldots, Y^{(t+T_f)}\right] = f\left(X^{(t-1)}, \ldots, X^{(t-T_b)}; \beta^{(t-1)}\right) \tag{4}$$

where T_f represents the number of time steps in the future that we are predicting for (i.e. look-forward period), T_b is the lookback period, and $Y^{(t)}$ represents a vector of predicted usage at all the stations at time step t . The objective is then to learn a function f to make predictions of usage in the future, based on past observations. In this study, the function f takes on the form of a spatiotemporal graph neural network (ST-GNN). The details of the ST-GNN are elaborated on in Section 3.2.

3.2. Neural Network Architecture

The neural network architecture used to predict the change in usage as a result of the addition of new stations is illustrated in Figure 1. This is the architecture used by Guo *et al*. [7], where the spatial correlations are modeled by a GCN which uses the propagation rule given by Eq. (3). The output of the GCN layers is then fed into gated recurrent units (GRU) [3], where the temporal correlations are modeled. We refer to this neural network architecture as GCN-GRU. The main difference in neural network used in this study is that the graph structure changes over time. As new bike sharing stations are opened, the weights of edges connecting to the new stations become non-zero. More specifically, when evaluating a training/test example, the graph $G^{(t)}$ used for the convolution is chosen based on the time t that the training/test example belongs to.

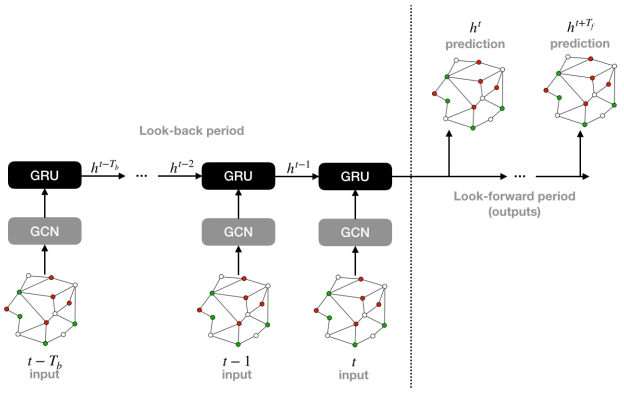

Figure 1: GCN-GRU neural network architecture used to predict the change in bike sharing usage as a result of the addition of new stations.

3.3. Implementation Details

The features used to train the network are given in Section 3.1, with one-hot encoding used to represent the categorical variables. In order to train the network, we employ a sliding window with a given lookback period T_b and look-forward period T_f, starting from the first day of the study period (i.e. Jan 1st, 2016) and ending when the sliding window reaches the end of the training set (i.e. Aug 31st, 2016). The resulting feature tensor $X \in \mathbb{R}^{|V| \times T_b \times d}$. The corresponding target matrix is then $Y \in \mathbb{R}^{N \times T_f}$. We train and evaluate the performance of the model using the Root Mean Square Error (RMSE) as the metric:

$$RMSE = \sqrt{\frac{1}{N \times T_f} \sum_{i}^{N} \sum_{t}^{T_f} \left(Y_{it} - \hat{Y}_{it} \right)^2} \qquad (5)$$

where Y_{it} and $\hat{Y}_{it}$ are the recorded and predicted number of trips starting from station i during hour t respectively.

The optimized set of hyperparameters was identified by first specifying a discrete set of hyperparameters, then performing grid search to find the set with the best validation RMSE score. The hyperparameters search space and the optimized set of hyperparameters, is given in Table 1.

During the training of the neural network, overfitting was avoiding through early stopping. This was implemented by evaluating the trained model on the validation set at the end of every epoch. If the validation RMSE does not decrease for more than 10 consecutive epochs, the training is stopped, and the best model is taken to be the model with the lowest validation RMSE score.

Table 1: Hyperparameter space tested and optimal hyperparameters for GCN-GRU

Hyperparameter	**Values Tested**	**Optimal Value**
Lookback period, T_b	-	12
Distance threshold for building the graph / km	0.3, 0.5, 1.0	0.3
Learning rate	0.0003, 0.001	0.0003
Number of GCN hidden layers	-	2
Dimension of GCN hidden layers	16, 32	16
Number of GRU hidden layers	-	2
Dimension of GRU hidden layers	32, 64	32
Dropout probability	0.1, 0.2	0.2

4. Experimental Results

4.1. Data

The datasets used in this study is the Citi Bike Trip Histories Data [4] and the National Oceanic and Atmospheric Administration weather data for New York Central Park area [12] for Jan 1st, 2016 to Dec 31st, 2016. Based on the distribution of when new stations were added to the system, we implement the data split as such: Jan 1st, 2016 to Aug 31st, 2016 for training, Sep 1st, 2016 to Sep 15th, 2016 for validation, and Sep 16th, 2016 to Dec 31st, 2016 for testing. This ensures that the validation and test sets have roughly the same number of new stations added to the network. An exploratory analysis of the data is done in Appendix A.

4.2. Hyperparameter Optimization

The optimal hyperparameters for the GCN-GRU network are listed in Table 1. The corresponding RMSE on the training and validation set when using this set of hyperparameters is 3.23 and 4.81 respectively.

4.3. Comparison with Benchmark Models

The performance of the GCN-GRU network is then compared to two other benchmark models:

- GCN-GRU$_{\text{initial}}$: GCN-GRU with the initial graph
- GCN-GRU$_{\text{final}}$: GCN-GRU with the final graph

The GCN-GRU$_{\text{initial}}$ and GCN-GRU$_{\text{final}}$ models are the same as the GCN-GRU model, except that the graph used in the GCN layer is kept the same throughout the training, validation and testing process. More specifically, in the case of the GCN-GRU$_{\text{initial}}$ model, the fixed graph used is the graph containing all the stations (old and new) as nodes,

Table 2: Comparison of models' RMSE

	Training	Validation	Test
GCN-GRU	3.23	4.81	3.94
GCN-GRU$_{\text{initial}}$	3.26	4.89	4.09
GCN-GRU$_{\text{final}}$	3.26	4.76	3.90

but only with edges between the old stations, while in the case of the GCN-GRU$_{\text{final}}$ model, the graph contains edges between the new stations as well.

The corresponding RMSE scores on the training, validation and test sets are listed in Table 2. From Table 2, we see that the GCN-GRU model performs better than the GCN-GRU$_{\text{initial}}$ model on the test set, with an RMSE loss of 3.94 compared to 4.09. The GCN-GRU$_{\text{final}}$ model performed marginally better than the GCN-GRU model, with an RMSE loss of 3.90 compared to 3.94. However, the difference is relatively small, and the GCN-GRU model can be said to provide favorable predictions in some cases compared to the GCN-GRU$_{\text{final}}$ model, as will be explained in Section 4.4.

4.4. Analysis and Discussion

Using the trained GCN-GRU network, we compare the predictions of the hourly number of bike trips starting from several stations (*i.e.* hourly usage) with their corresponding historical observations (denoted "labels") in Figures 2-3. In each of the figures, we plot the predicted hourly usage (blue line) and the labels (green line) over a period of a week. Here, we consider stations which opened on dates in the test set. For the plots of other stations, we refer the reader to Appendix B.

Figure 2 shows the hourly usage for the station named "Bergen St & Smith St" located in Brooklyn, which opened on Sep 19, 2016. The blue line represents the predictions made by the GCN-GRU model, while the red line denotes the predictions made by the GCN-GRU$_{\text{initial}}$ model. We observe that the predicted usage before the opening of the station is close to zero as expected, before closely following the labels curve after the opening of the station. On the other hand, the GCN-GRU$_{\text{initial}}$ model predicts only a constant usage over the whole week, showing that it was unable to model the usage the hourly usage of new stations that opened in the test set. Hence, this shows the importance of using a graph that has updated edges from the new stations.

In Figure 3, we consider another station which opened in the test set on Sep 21, 2016, named "E 74 St & 1 Ave". The blue line shows the predictions from the GCN-GRU model, while the red line denotes the predictions from the GCN-GRU$_{\text{final}}$ model. We see that the GCN-GRU model initially predicts close to zero hourly usage at the start of the week. However, two days before the opening of the station, the predicted usage started increasing, illustrating some limitations of the model. On the other hand, the GCN-GRU$_{\text{final}}$ model predicted that the hourly usage for the days prior to the opening of the station was significantly larger than zero. Hence, compared to the predictions of the GCN-GRU model, the GCN-GRU$_{\text{final}}$ model had worse predictions in some cases, even though it had a smaller RMSE loss. This example shows that there could be added benefits to using a dynamic graph with a GCN, where edges are only added when new stations come into service.

In an attempt to better understand the shortcomings of

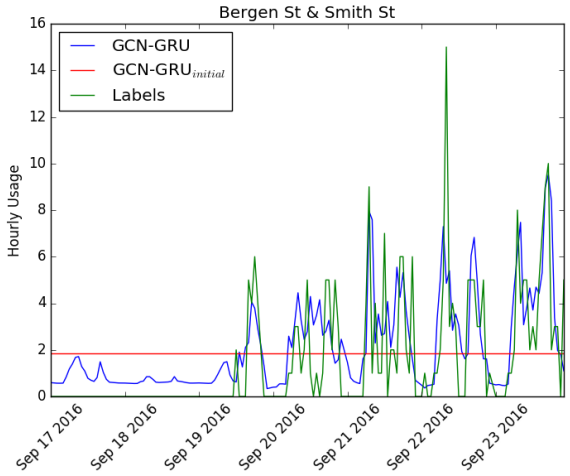

Figure 2: Predicted values by (a) GCN-GRU and (b) GCN-GRU$_{\text{initial}}$ and actual labels for number of bike trips per hour starting at one of the stations which opened on Sep 19, 2016.

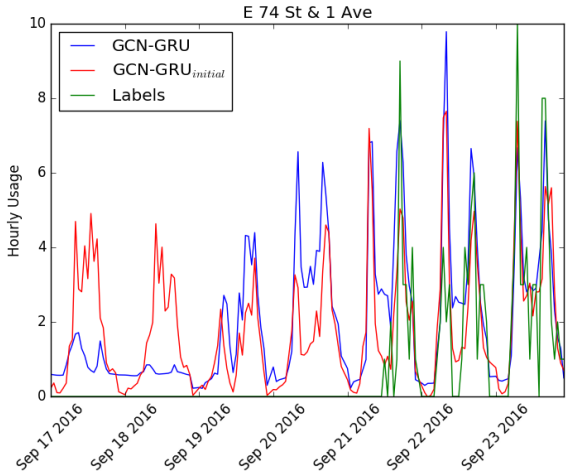

Figure 3: Predicted values by (a) GCN-GRU and (b) GCN-GRU$_{\text{final}}$ and actual labels for number of bike trips per hour starting at one of the stations which opened on Sep 21, 2016.

the GCN-GRU model, we considered the case when the distance threshold κ for building the graph was set to zero. When this happens, there are no edges in any of the graphs $G^{(t)}$. According to Eq. (3), the propagation rule would then become that of a fully-connected layer (FCL). It turns out that this model, which is made up of FCLs and GRU layers (*i.e.* just an RNN), had an RMSE score of 2.39, 3.50 and 2.94 for the training, validation and test sets respectively, which is better than those of the GCN-GRU model. This indicates that the distance-based graph used in the GCN-GRU model is not capturing useful spatial correlations, and shows the importance of the graph structure in GCN.

In view of this, we constructed a GCN-GRU model with a dynamic graph, but using demand correlation instead. Here, edges between nodes were added if the correlation between the two stations' demands was greater than 0.75. The resulting RMSE scores for the training, validation and test set were 2.64, 3.72 and 3.30 respectively. This represents an improvement over the scores of the GCN-GRU model with distance-based graphs, hence showing the importance of the graph structure when capturing spatial correlation. However, it is still worse than the RNN model, which could be due to the fact that the problem we are attempted to solve is not suitable for the GCN framework.

5. Conclusion

We explored the use of a GCN in combination with gate recurrent units for predicting the hourly number of bike trips starting from all the stations in a bike-sharing network, including new stations that were added to the network during and after the training phase of the neural network. We showed that the GCN-GRU network was able to learn and predict the one-hour-ahead hourly usage for both old and new stations, by using a dynamic graph of the bike-sharing network with edges added when new stations come into service. With more data, it would be possible to consider more aggregated time intervals (e.g. 1 week or 1 month) and make even more long-term predictions of usage.

However, there is still room for improvement in terms of prediction accuracy. First, the graph structure, or criterion used to construct the graph, is very important in determining the performance of the graph convolutional network. Second, the use of dynamic graphs in a graph convolutional network is still an actively studied area. As new and improved methods are proposed, the performance of graphical convolutional networks for predicting usage as new stations are added to the bike-sharing network is likely to improve.

References

[1] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[2] 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*, pages 841–852. ACM, 2016.

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[4] Citi Bike NYC. Citi Bike system data. `https://www.citibikenyc.com/system-data`. Accessed 10/23/2019.

[5] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[6] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *2019 AAAI Conference on Artificial Intelligence (AAAI'19)*, 2019.

[7] R. Guo, Z. Jiang, J. Huang, J. Tao, C. Wang, J. Li, and L. Chen. Bikenet: Accurate bike demand prediction using graph neural networks for station rebalancing.

[8] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[9] 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*, page 33. ACM, 2015.

[10] 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*, 97:258–276, 2018.

[11] J. Liu, L. Sun, W. Chen, and H. Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM, 2016.

[12] National Weather Service Forecast Office. NOAA online weather data. `https://w2.weather.gov/climate/xmacis.php?wfo=okx`. Accessed 10/23/2019.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[14] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[15] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng. Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1227–1235. ACM, 2019.

[16] Z. Yang, J. Hu, Y. Shu, P. Cheng, J. Chen, and T. Moscibroda. Mobility modeling and prediction in bike-sharing systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 165–178. ACM, 2016.

[17] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 2018.

[18] J. Zhang, Y. Zheng, and D. Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[19] 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.

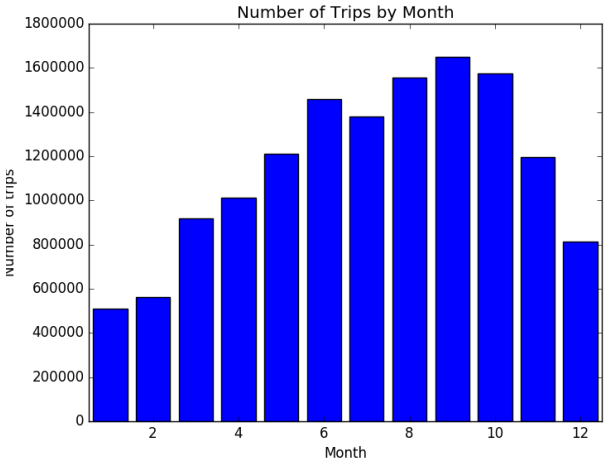

Figure 4: Number of Citi Bike trips by month in 2016.

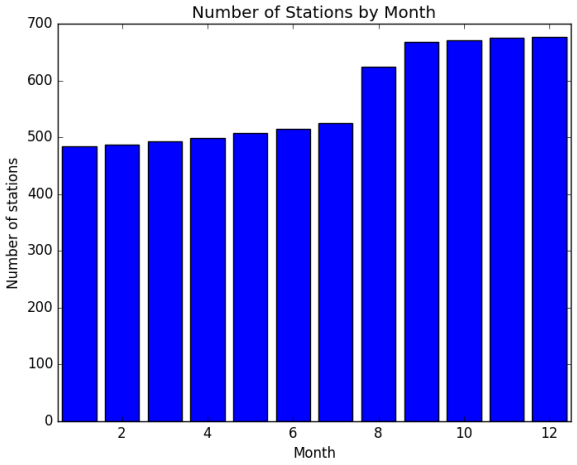

Figure 5: Number of stations in service in the Citi Bike network by month in 2016.

A. Exploratory Data Analysis

Before commencing training on our dataset to predict the effects of opening a new station, we first conduct exploratory data analysis to better understand the data we are working with. Specifically, we would like to understand the change of number of trips and number of stations, as well as where these new stations are, over the 12 months of 2016.

The number of Citi Bike trips by month is plotted in Figure 4. Based on the distribution in Figure 4, it is clear that there are more trips in the summer and fall, when the weather is optimal for outdoor activity.

The number of stations in service per month in 2016 is shown in Figure 5. Each month, there are more stations than the previous month. The number of stations drastically increased in Aug and Sep, with the opening of stations in Brooklyn and Uptown.

Figure 6 illustrates the locations of all the stations (both old and new) in the network in 2016. The old stations are represented by the orange markers, while the blue markers denote new stations which opened in 2016. The new stations which opened in 2016, as shown in Figure 6, were clearly concentrated in Uptown (north end of Upper East Side and Upper West Side), as well as in New Jersey and in Brooklyn. The number of stations that opened in 2016 Midtown, Downtown, or Queens was minimal.

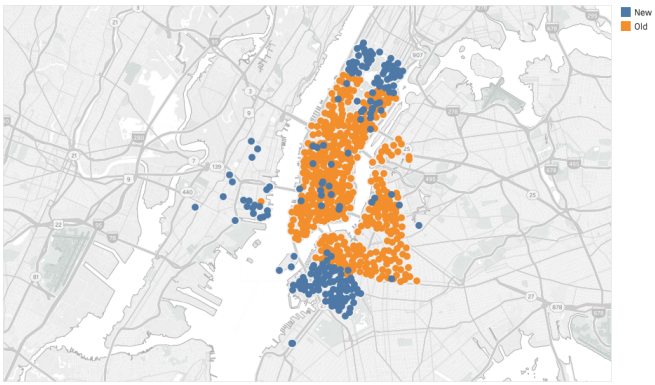

Figure 6: Locations of Citi Bike stations, with different colors denoting whether a given station was opened in 2016.

B. Supplementary Diagrams

The plot in Figure 7 shows the hourly usage for an old station (i.e. existing before 2016) named "W 17 St & 8 Ave" during a randomly chosen week in the test set. We see that the predictions by the GCN-GRU model are able to capture the day-night patterns, along with some of the morning and evening peak patterns. However, the predictions were slightly off in terms of predicting the magnitudes of some of the peaks.

In Figure 8, we plot the hourly usage for a station named "W 27 St & 10 Ave", which opened during the training period, to see how well the GCN-GRU model is able to predict for a new station which opened partway in the training set. Figure 6 shows that the predictions for this station during a week in the test set are generally quite good, with the prediction curve following the labels curve quite closely.

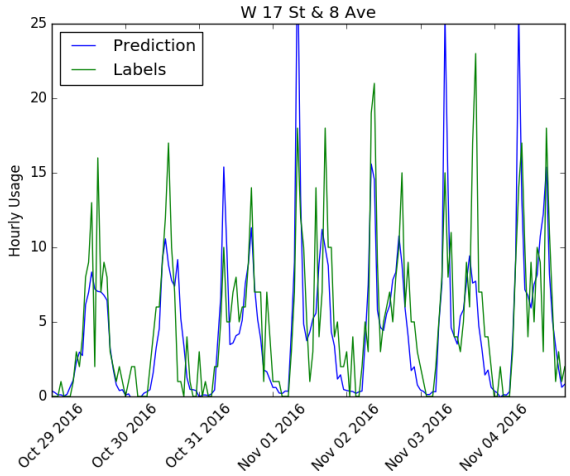

Figure 7: Predicted values and actual labels for number of bike trips starting at one of the old stations existing before 2016.

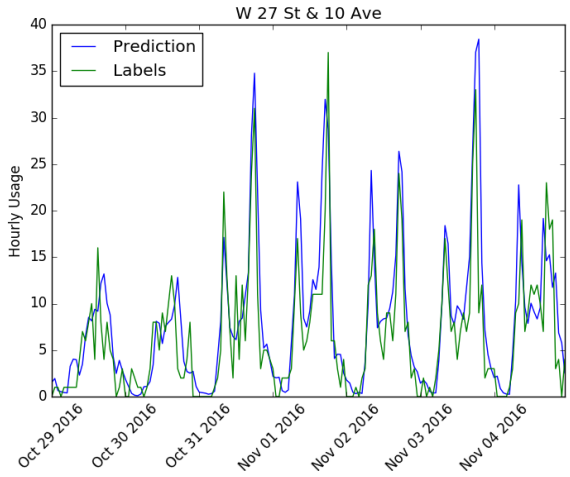

Figure 8: Predicted values and actual labels for number of bike trips starting at one of the new stations which opened during the training period.