
Time Series Analysis on Transportation Usage and COVID-19

Cora Hyun Jung, Emmy Phung, Sujeong Cha
New York University
60 5th Ave, New York, NY 10011
{hj1399, mtp363, sjc433}@nyu.edu

Abstract

Since the coronavirus 2019 (COVID-19) outbreak in New York City, people have been advised to keep social-distancing and avoid public transportation, which results in a steep change in the mobility patterns. This project investigates the level of impacts that COVID-19 has had on four means of transportation in NYC, including subway, taxi, for-hire vehicles (FHV, e.g. Uber/Lyft), and Citi Bike. In order to tackle this problem, we suggest two approaches: 1) analyze cross-correlation between COVID-19 and the ridership and 2) build a pre-COVID model to predict ridership during the post-COVID era to see the differences. The results from both approaches are consistent, demonstrating that all of the transportations have been affected by COVID-19. According to our findings, the FHV usage pattern has changed the most drastically and Citi Bike ridership has been the least affected by the pandemic.

1 Introduction

The outbreak of COVID-19 began in the US on February 29th, 2020, when the first positive case was reported. Because of its contagion and severe health consequences, populated areas such as New York City immediately became high-risk areas. To mitigate the spread of COVID-19, New York City, together with a majority of other cities and countries in the world, has encouraged and at times enforced social distancing as well as self-quarantine. Governor Cuomo's order was announced to have everyone but essential workers stay home and for schools and most businesses to be closed [1]. This has vastly changed the mobility patterns in the city, tremendously lowering the demand for public transportation or shared vehicles while promoting no mobility or personal ones. Since our daily lives are heavily affected by means of transportation, it's essential to investigate the relationship between mobility trends and COVID-19.

Our goal is to investigate the severity of the mobility pattern change across four means of transportation in NYC during the pandemic, which includes subway turnstile usage, taxi pickup counts, FHV pickup counts, and Citi Bike ridership. In order to tackle this problem, we suggest two approaches: 1) analyze cross-correlation between COVID-19 data and the ridership data during the post-COVID era and 2) build a pre-COVID model that captures the trend and seasonality of ridership then use it to predict ridership during post-COVID-19 to see the differences. For the second approach, we build five different time series models, including ARMA, Facebook Prophet, Gaussian Process (GP), Kalman Filtering (KF), and RNN. The results show that among the four modes of transportation, FHV is most heavily affected by COVID-19, whereas Citi Bike has been least affected.

2 Related Work

Since the outbreak, several related research works have been conducted by researchers as well as stakeholders, such as transportation companies. Fathi-Kazerooni, et al. [2] apply an autoregressive

integrated moving average (ARIMA) model to estimate the COVID-19 trend and forecast the dates of minimum cases and deaths, as well as a long short-term memory (LSTM) model to demonstrate the correlation between NYC subway usage and COVID-19 death counts.

Another inspiration for applying deep learning into tackling time series modeling comes from research by Uber. Zhu, et al. [3] use an LSTM model with an autoencoder to forecast future rideshare demand. According to the authors, LSTM can function as an autoencoder to create embeddings of the sequential data and extract time-dependent features.

These ideas motivate us to first tackle the issue with traditional time series models, such as ARMA, Kalman Filtering, Gaussian Process, etc. In addition, we explore the use of LSTM-based model architecture for point-wise prediction and utilize the predicted inputs to perform long-term or sequential prediction. We also attempt to apply the Prophet model developed by Facebook because it's simple to build and performs relatively well.

3 Problem Formulation and Algorithms

Observing tremendous changes in the mobility trend in New York City during the pandemic, we are motivated to develop a quantitative method to measure the impacts of the pandemic on transportation patterns, in which the four selected subjects of research are subway, taxi, FHV, and Citi Bike. In particular, we are interested in 1) estimating the correlation between the severity of COVID-19 and transportation usage and 2) comparing such impacts across these four means of transportation.

3.1 Approach 1: Cross Correlation Function

To answer our first research question, we investigate the Cross Correlation Function (CCF) between COVID-19 and each transportation. CCF measures the linear predictability of one time series from another. If the value is positive, the two series are moving in the same direction at that time lag, and vice versa. This approach allows us to see a general picture of the directions of relationships and also gives insights about the specific lag when the momentum starts to change its direction.

3.2 Approach 2: Model Comparison on Pre-COVID and Post-COVID Test Sets

In the second approach, we hypothesize that the steep change in transportation patterns during the pandemic will prevent a model trained on pre-COVID data from performing well on post-COVID data. In particular, we want to develop a univariate time series model that can capture the seasonality and trend of pre-COVID ridership across different means of transportation. We then test this model on two test sets, pre-COVID and post-COVID, and compare their prediction errors.

Our chosen metric is the mean squared error (MSE), and the difference in MSE (in basis points) between the two test sets is calculated to indicate the level of impacts that the pandemic has on the mobility patterns of these vehicles. Our rationale behind this experiment setup is that if a model that performs well on pre-COVID test set fails to give good predictions on post-COVID test set, there must be a fundamental change in NYC's mobility patterns after the COVID-19 outbreak. On the other hand, if the difference in MSE is small, meaning that the model performs similarly on the two test sets, we can conclude that the pandemic does not cause any significant consequences on transportation patterns in NYC.

Since time series modeling on non-stationary data is not an easy task, we do not rely on one model but aggregate results across five different time series models to derive conclusions for this approach.

3.2.1 Model Explanations

For the second approach, we build a total of five different models, including ARMA, Facebook Prophet, Gaussian Process, Kalman Filtering, and RNN. We will give further explanation below and discuss implementation details in section 4.

ARMA Autoregressive moving average (ARMA) model consists of two processes: autoregressive (AR) and moving average (MA). It uses two hyper parameters, p and q , where p is the order of the AR process and q is the order of MA process. The ARMA equation can be written as follows:

$$ARMA(p, q) : Y_t = c + \sum_{i=1}^p \phi_i Y_{t-i} + \sum_{i=1}^q \theta_i \epsilon_{t-i} + \epsilon_t$$

where Y_t represents the series itself, c represents a constant, p defines the number of lags to regress against Y_t , ϕ_i is the coefficient of the i lag of the series, q defines the number of past error terms, θ_i is the corresponding coefficient of ϵ_{t-i} , and $\epsilon_{t-q}, \dots, \epsilon_t$ are white error noise terms.

Facebook Prophet Prophet is an open-sourced library developed by Facebook, which provides an automated forecasting model to make time-series analysis more accessible to non-technical users [4]. It’s also customizable, with options to control uncertainty, trend and holiday effects. The model itself is similar to the general additive model with the following prediction equation.

$$y(t) = g(t) + s(t) + h(t) + X(t)\beta + \epsilon_t$$

where $g(t)$ signifies non-periodic trend, $s(t)$ is periodic, seasonal trend (weekly, yearly seasonality), $X(t)$ is regressors portion, and $h(t)$ is holiday effects. It appears to resemble an ARIMA prediction function, but the main difference is that the residual in the Prophet model is not assumed to have autocorrelation. While an ARMA model can only handle stationary data, Prophet can tackle non-stationary cases. Prophet is built with its backend in STAN, a probabilistic coding language, which allows it to have many advantages provided by Bayesian statistics, including seasonality, easy application of domain knowledge, and confidence intervals to estimate risks [4].

Gaussian Process The next algorithm is Gaussian Process (GP). GP is a collection of random variables where every finite linear combination of them forms a multivariate Gaussian distribution. Just as any Gaussian distribution is defined by its mean and covariance, GP can be fully specified by a mean and covariance function. A standard GP model assumes a stationary covariance. However, we can use a non-stationary kernel inside a covariance function to capture non-stationarity in the data. Since all of our datasets have high seasonality and non-zero means, we can benefit from a GP model with non-stationary kernels. There are multiple Python-based packages for GP such as Scikit-learn, GPytorch, and GPflow, but we will use Scikit-learn for its usability and precise documentation.

Kalman Filtering The fourth model we experiment with is Kalman Filtering (KF). Under KF, all observations are assumed to have measurement noise. The main goal of this algorithm is to find a sequence of latent states hidden behind the noisy observations. KF works in two steps - filtering and smoothing. During filtering, the algorithm predicts the next latent state based on the previous latent state and the current observation (evidence). During smoothing, we propagate information backward in time using the current filtered state and the filtered and smoothed state from the next timestamp. Parameters, such as transition matrix and observation matrix, are updated by applying the expectation-maximization (EM) algorithm. KF is “suitable for application to a stationary or a non-stationary time series” [5]. Snyder, et al. also note that “it can be used on seasonal time series where the associated state space model may not satisfy the traditional observability condition,” [5] which suggests that this model is applicable to our non-stationary transportation data.

RNN The last model that we want to explore is the recurrent neural network (RNN), especially the LSTM model which has shown great potential in modeling time series data. Generally speaking, RNN works well with sequential data because it has a looping mechanism that allows information to be passed from one state to another, creating a form of “memory” and enabling the model to learn a sequence of inputs instead of treating them as unconnected data points. One limitation of RNN is that it faces the infamous issue of vanishing gradients, which prevents the model from learning a long sequence of inputs and causing a so-called “short-term memory” [6]. This limitation of RNN has been resolved by its modified version, LSTM. LSTM uses mechanisms called “gates”, which are different tensor operations that are trained to learn what information to remove or pass onto the next hidden states [6]. These gates mitigate the problem of vanishing gradients and enable LSTM to learn long-term dependencies within an input sequence. Furthermore, LSTM has recently become more popular in solving time series problems because of its automatic feature extraction and flexibility in incorporating exogenous variables [6]. For this project, we focus on a simple LSTM-based model trained on fixed-size sequences of input data and a single point prediction is performed at a time. The preprocessing of input data and model architecture will be further discussed in section 4.

131 4 Data Overview

132 4.1 Data Collection and Preprocessing

133 **Subway** The turnstile usage data of the New York City subway is web scraped from the website
134 of Metropolitan Transport Authority of New York City (MTA) [7]. Preprocessing this dataset is
135 particularly challenging. The MTA publishes turnstile data weekly, which contains about 200,000
136 rows, each representing an observation at a moment in time for a specific turnstile device. The
137 readings are logged every four hours.

138 The most important note for pre-processing is that the entry counts are cumulative. To calculate the
139 net entries for a certain period of time, we first sort the compiled dataset by unique logging devices
140 then compute the difference between two consecutive logs. The aggregated data is then grouped
141 by days so that each row represents daily turnstile entry counts. Detailed implementation is further
142 explained in Appendix A.

143 **Taxi/For-Hire Vehicles** Taxi (Yellow Taxi + Green Taxi) and FHV (For-Hire + High Volume
144 For-Hire) datasets are web-scraped from NYC Taxi Limousine Commission website [8]. Because
145 each row represents a single pick-up, the number of rows is approximately 9.3 million per month (for
146 Yellow Taxi only), which is time-consuming to download and compile all data over five years. The
147 compiled pickup records are merged and preprocessed to give the daily pickup counts

148 **Citi Bike** The Citi Bike ridership data is downloaded from the Citi Bike website [9]. Similar to taxi
149 data, Citi Bike data is reported per month and each row of a dataset represents a single trip, which
150 adds up to about 2 million rows per monthly data file. The data is compiled and grouped by days to
151 create the final dataset containing daily bike ridership.

152 **COVID-19** The COVID-19 dataset is downloaded from the website of the NYC Department of
153 Health (DOH) [10], which provides the statistics of the confirmed COVID-19 cases and deaths. The
154 dataset is preprocessed to give the daily case counts.

155 4.2 Exploratory Data Analysis

156 The subway and taxi data show strong seasonality and periodicity (Appendix B Figure 2, 3). Since
157 the start of the pandemic, the usage of both transportations has significantly dropped but seasonality
158 remains. While the Citi Bike ridership has declined right after COVID-19 hit, it has quickly recovered
159 to its normal patterns or has had even higher ridership count (Appendix B Figure 4). We also find that
160 while both traditional taxi and FHV usage has dropped significantly after COVID-19 hit, FHV plot
161 displays a quicker recovery (Appendix B Figure 3). The COVID-19 data show the large prevalence of
162 COVID-19 in April (Appendix B Figure 1). Thus, to test the impact of COVID-19 on transportation
163 means the best, it would be appropriate to include April in post-COVID transportation test data.

164 4.3 Data Split

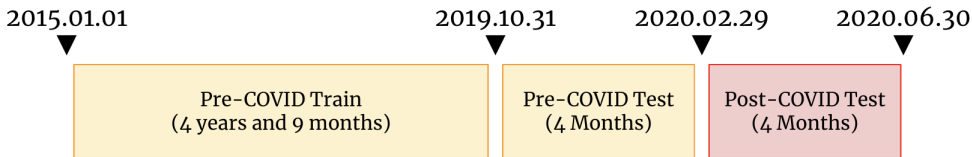


Figure 1: Data splitting for all four transportation datasets

165 All four transportation datasets range from January 1st, 2015 to June 30th, 2020, and are split in the
166 same way for comparison purposes (Figure 1). First, we divide a dataset into two, pre-COVID and
167 post-COVID, using March 1st, 2020 as the splitting point. We then split pre-COVID data into train
168 and test sets, in which the training data contains roughly five years of ridership data and the test data
169 contains four months. The post-COVID dataset ranges from March 1st, 2020 to June 30th, 2020,

170 which is about the same length as our pre-COVID test set. All of the datasets are standardized for
171 valid comparison across different datasets.

172 5 Methodology and Results

173 5.1 Approach 1: Cross Correlation Function

174 We compute CCF on normalized data as discussed in section 4.3. Since our aim for Approach
175 1 is to generate a high-level understanding of long-term relationships, seasonality in individual
176 datasets needs to be removed. Thus, we decompose all the datasets, including COVID, using
177 statsmodels.seasonal_decompose, to extract trend and seasonality, then estimate CCF on the trend
178 component.

179 Figure 2 shows the cross correlation between COVID-19 daily cases and each transportation means.
180 From these plots, we can draw the following conclusions. First, COVID-19 affects for-hire vehicles
181 and subway in an extremely similar fashion. The CCFs for both transportations (Subplot 2, 3) start
182 to decrease from April 6th, 2020 where the COVID-19 daily cases peak and then plateau at around
183 June 18th, 2020. Moreover, the signs for these CCFs flip to negative, meaning that the usage for
184 for-hire vehicle and subway increases while COVID-19 infection decreases around May 6th, 2020. It
185 is interesting that this observation corresponds to one month before the official Phase 1 reopening of
186 the city (June 8th, 2020).

187 Second, we notice that the tail for FHV plot (Subplot 2) is a bit higher than that of taxi (Subplot
188 1), which indicates that FHV shows a sign of quicker recovery than taxi. This is understandable
189 because the majority demand for taxi comes from tourists while most local residents use Uber and
190 Lyft. Furthermore, after the first and second phases of city reopening, New Yorkers feel safer to
191 commute with Uber and Lyft compared to taxi as they can easily book a ride and communicate with
192 drivers via mobile apps.

193 Lastly, unlike taxi and subway, Citi Bike (Subplot 4) has an upward sloping CCF even after COVID-19
194 peaks, which implies a positive relationship between COVID-19 and Citi Bike during the pandemic.
195 Combined with the raw plot for Citi Bike, we can conclude that the pandemic even makes Citi Bike
196 more popular.

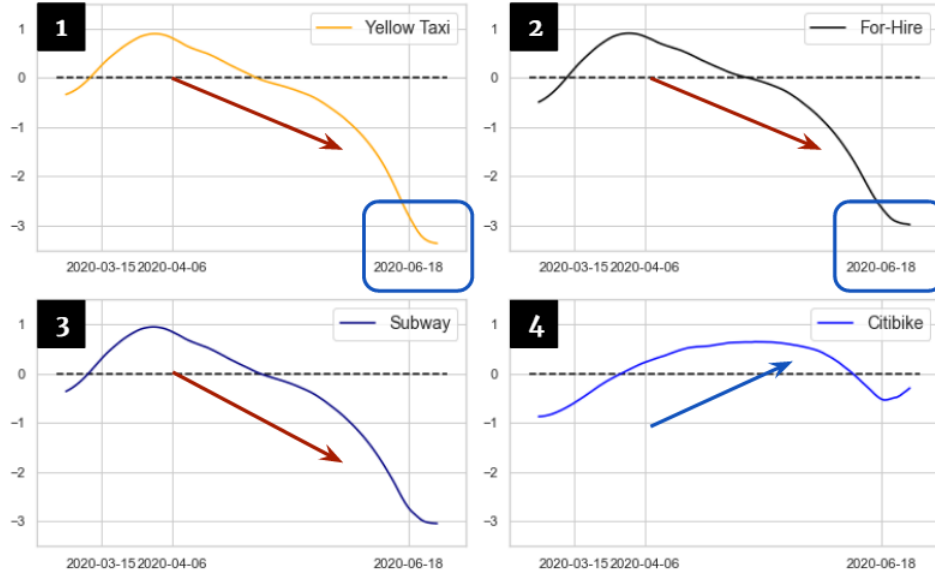


Figure 2: CCF between daily COVID-19 cases and transportation usage

Table 1: Result summary across five models and four datasets

Model	Taxi			For-Hire Vehicles			Subway			Citi Bike		
	Pre	Post	Diff	Pre	Post	Diff	Pre	Post	Diff	Pre	Post	Diff
ARMA	0.15	3.87	24.64	0.22	7.97	34.58	1.63	11.72	6.19	2.47	3.53	0.43
GP	0.10	3.35	31.54	0.20	8.71	43.09	0.35	11.16	30.88	0.36	2.60	6.16
Prophet	0.08	3.81	46.27	0.14	7.76	54.34	0.27	12.00	43.45	0.47	2.34	3.98
RNN LT	0.10	4.67	44.65	0.16	5.28	31.75	0.42	11.52	26.17	1.85	1.21	-0.35
RNN PW	0.03	1.24	36.79	0.07	0.51	6.42	0.25	7.75	30.23	0.24	0.46	0.94

* *Pre* refers to Pre-COVID MSE, *Post* refers to Post-COVID MSE, and the difference is measured in %

5.2 Approach 2: Model Comparison on Pre-COVID and Post-COVID Test Sets

ARMA The first model we experiment with is ARMA. The Dickey Fuller test is used to check stationarity, which shows that our datasets are not stationary. Thus, the data is detrended and deseasonalized using linear regression and seasonal_decompose package from statsmodel (Appendix B Figure 5). To estimate the ARMA order (p, q), we plot a partial autocorrelation function (PACF) to find p and an autocorrelation function (ACF) to find q. In addition to ACF and PACF, the grid search is performed to find the p and q that minimize the Akaike’s Information Criterion (AIC), an in-sample statistic. Then, using additive prediction function below, we obtain the residual by subtracting trend and seasonality from the data and then fit an ARMA model on the residual.

$$Y(t) = T(t) + S(t) + e(t)$$

where $T(t)$ signifies the non-periodical trend, $S(t)$ is seasonality, and $e(t)$ is the residual.

ARMA model is performed on all four transportation data, and the impact of COVID-19 is measured by taking the percent difference between pre-COVID MSE and post-COVID MSE (Table 1). The results show that among four transportations, FHV exhibits the highest difference in MSE, meaning that it is the most severely affected by COVID-19. Appendix B Figure 6 depicts the ARMA prediction result for the taxi data. It captures patterns fairly well during pre-COVID era; however, for post-COVID ridership, the prediction differs from the truth by a large margin.

Facebook Prophet Using Prophet is fairly simple thanks to the pre-built methods in the package. We build a Prophet model and set the periods to four to make a four-month forecast. Appendix B Figure 8 shows that the model can capture both non-periodic trend and seasonality of subway data including daily, weekly, and yearly patterns. While the model well captures the sinusoidal patterns of subway ridership during the pre-COVID era, predictions for post-COVID ridership are highly inaccurate (Appendix B Figure 7). The results in Table 1 show that the FHV also exhibits the highest difference in MSE, demonstrating that it’s the most severely affected by COVID-19.

Gaussian Process Across all transportations, a combination of Radial Basis Function (RBF), Exponential Sine Squared, and Constant Kernel shows a solid performance. However, the periodicity parameter for Exponential Sine Squared Kernel needs to be adjusted according to each transportation’s seasonality. The periodicity parameter of seven was chosen for taxi, FHV and subway due to the strong weekly seasonality, and 365 was chosen for Citi Bike as it displays the strong yearly seasonality. Moreover, all datasets contain a linear long-term trend. Thus, multiplying Exponential Sine Squared Kernel by DotProduct Kernel well captures the linear trends while preserving the seasonal cycle. Appendix B Figure 9 exhibits the predictions and the ground truth for for-hire vehicles. The red prediction line clearly depicts the periodic movement and the positive linear trend. According to Table 1, FHV shows the largest difference between the MSE for pre-Covid and post-Covid, which confirms that it is the most severely affected transportation by COVID-19.

RNN As mentioned in section 3.2.1, LSTM is capable of learning long-term and short-term dependencies in time series data. We first preprocess the data to create suitable training inputs for RNN. Using a sliding window of size N, we generate training samples that consist of one data sequence $(x_1 \cdots x_N)$ and one label (x_{N+1}) . The size of our sliding window is determined by the number of lags found in Dickey Fuller test’s results, which ranges from 22 to 23 days depending on the

data. Next, we build a baseline model with one layer of LSTM, one dropout layer for regularization, and one fully connected layer. We then train this model with Adam Optimizer (initial learning rate of 0.001) and use MSE as our loss function. We also attempt to increase the number of LSTM layers, but we find that a shallower network actually performs better on the validation set due to the small data size. Similarly, we perform the hyper-parameter tuning in search for the right number of hidden units using the range suggested in Uber’s research paper (32, 64, and 128) and find 64 to yield the best result [3].

For future forecasting, we experiment with both point-wise prediction (PW) and long-term prediction (LT). Point-wise approach entails that we constantly feed the model the actual data up to the last time step as inputs and have it predict the next data point. In the other approach, we create a loop to add newly predicted points to the input sequence and generate continuous predictions, which we refer to as long-term prediction (LT). As shown in Table 1, RNN-PW performs much better than RNN-LT on both pre-COVID and post-COVID test sets because it only has to perform a single point prediction based on actual input data while RNN-LT makes prediction based on previously predicted points. Based on both RNN-PW and RNN-LT, traditional taxi is the most severely affected by COVID-19 and Citi Bike is the least affected (Appendix B Figure 10). This result is consistent with other models and will be further discussed in section 6.

Kalman Filtering All parameters for Kalman Filtering except the initial mean (set to zero) are set to the identity matrix to begin with. Then, the EM algorithm is applied to the training set with 10 iterations. Predictions for the pre-COVID and post-COVID test sets are calculated based on the fitted parameters. We first compute the mean and covariance prediction for each future timestamp and draw a random sample from a normal distribution with the corresponding mean and covariance.

Initially, Kalman Filter is applied to raw data without removing seasonal, non-stationary components. No matter how we train KF, the model fails to fit the training data, resulting in very high train and test MSE. The result for for-hire vehicles is shown in Appendix B Figure 11. In the second attempt, we decompose the datasets and focus on the residual for both training and validating the model just as in ARMA models. Appendix B Figure 12 displays the prediction results for for-hire vehicles using this approach. Even though we achieve a lower train and test MSE than the previous attempt, we can still see the high discrepancy between the actual training data and model’s fitted values.

After several experiments, we come to a conclusion that KF is not applicable to our dataset for two reasons. First, during training, KF tends to smooth out the variance coming from seasonal effects because the algorithm only focuses on finding a sequence of latent states. Second, during prediction, we need to randomly draw a sample from a Gaussian distribution, which leads to very high variance in prediction results. These issues become more problematic when we perform long-term prediction because the randomness accumulates.

6 Conclusion & Final Discussion

Since time series modeling on non-stationary data is a challenging task, we aggregated results from two approaches and four models to draw our conclusion. Figure 3 summarizes the level of impacts of COVID-19 on NYC’s transportation usage patterns across our four final models. According to all four, Citi Bike ridership is the least impacted one and has even gotten more popular during the pandemic. The largest change in pattern has been observed in taxi and FHV, followed by subway ridership. The results drawn from our two approaches are consistent and are in line with our exploratory data analysis. We also think that the results are reasonable. Since biking is an individualistic activity, the threat of COVID-19 is smaller for bikers. However, sharing the same car with someone or being in close proximity with other commuters, like taking a subway, may impose a much higher threat and has been widely discouraged.

Besides validating our key hypotheses, the project also gave us meaningful insights on the four time series models we have implemented. Overall, RNN-PW gives the lowest MSE on both pre-COVID and post-COVID test sets. However, we cannot conclude that RNN is the best performing model on time series data because its long-term prediction is not consistently better than the others. In this aspect, Prophet appears to be the best performing model due to its consistency across different datasets. Last but not least, ARMA and Gaussian Process tend to overfit to training data and have shown varying performances across different datasets.

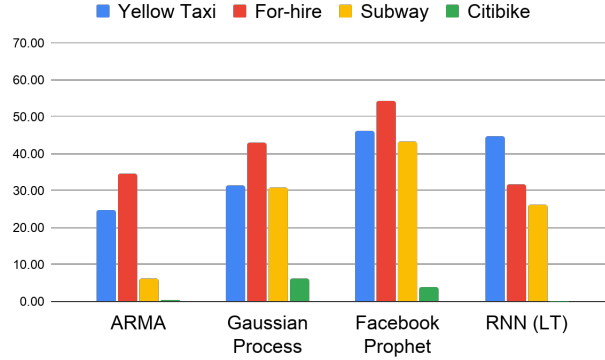


Figure 3: Impacts of COVID-19 on transportation usage (Δ MSE in %)

7 Future Directions

7.1 Forecast in Regression Format

For the scope of this project, we mainly focused on comparing the COVID-19 impact between various transportation means. Because the taxi data is available only up to June 30th, 2020, we had to use the same date range for the other datasets as well for valid comparison across different datasets. As a result, we did not have enough data to perform future predictions using regression analysis. However, as more data becomes available in the future, one possible extension of our project could be forecasting the ridership demand for the unseen future in a regression format. One possible hypothesis is that “If COVID-19 daily cases increase by X%, the total ridership for subway will decrease by Y%.” If the predictions do not consider the time horizon, then we can simply perform a linear (or polynomial) regression with the number of COVID-19 cases as an independent variable and the total number of ridership as a dependent variable. However, if we want to make a prediction for a specific time period, one way to tackle this problem can be Vector Auto Regression (VAR) - one of the most commonly used methods for multivariate time series forecasting [11]. Like the autoregressive model, the equation includes the variable’s lagged (past) values, the lagged values of the other variables in the model, and an error term [12].

7.2 Incorporating Holidays/Events

Another interesting extension is to incorporate the holidays and special event dates. As shown in Appendix B Figure 7, the prediction results show that the predictions are closely matched with the ground truth except for the holidays such as Thanksgiving, Christmas, and New Year’s Day. In the future, we can provide the list of holidays to Prophet to take this effect into account. Also, a paper from the U.S. Census Bureau [13] presents an example case - the authors use a retail dataset and incorporate holiday effects (Black Friday, Cyber Monday, etc) by treating holiday effects as one separate regression variable. However, incorporating holidays in non-regression models, such as Gaussian Process, needs to be investigated further.

References

- [1] New York State Government (2020). Governor Cuomo Signs the 'New York State on PAUSE' Executive Order. Retrieved December 15, 2020 from <https://www.governor.ny.gov/news/governor-cuomo-signs-new-york-state-pause-executive-order>
- [2] Fathi-Kazerooni, S., Rojas-Cessa, R., Dong, Z., Umpaichitra, V. (2020). Time Series Analysis and Correlation of Subway Turnstile Usage and COVID-19 Prevalence in New York City. 1-8. <https://arxiv.org/abs/2008.08156>
- [3] Zhu, L., Laptev, N. (2017). Deep and Confident Prediction for Time Series at Uber. 2017 IEEE International Conference on Data Mining Workshops (ICDMW). <https://doi.org/10.1109/icdmw.2017.19>
- [4] Taylor, S. J., Letham, B. (2017, September 27). Forecasting at scale. <https://peerj.com/preprints/3190/>
- [5] Snyder, R. D., Forbes, C. S. (2003). Reconstructing the Kalman Filter for Stationary and Non Stationary Time Series. Studies in Nonlinear Dynamics Econometrics, 7(2). <https://doi.org/10.2202/1558-3708.1087>
- [6] Phi, M. (2020, June 28). Illustrated Guide to Recurrent Neural Networks. <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9>
- [7] Turnstile Data. Retrieved September 30, 2020, from <http://web.mta.info/developers/turnstile.html>
- [8] NYC Taxi and Limousine Commission (n.d.). TLC Trip Record Data. Retrieved September 30, 2020, from <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>
- [9] Citi Bike (n.d.). Citi Bike System Data. Retrieved September 30, 2020, from <https://www.citibikenyc.com/system-data>
- [10] Nychealth. [nychealth/coronavirus-data](https://github.com/nychealth/coronavirus-data). Retrieved September 30, 2020, from <https://github.com/nychealth/coronavirus-data>
- [11] SinghAn, A. (2020, May 7). Multivariate Time Series: Vector Auto Regression (VAR). Retrieved December 2, 2020, from <https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes/>
- [12] Vector autoregression (2020, December 2). In Wikipedia. https://en.wikipedia.org/wiki/Vector_autoregression
- [13] Washington: US Census Bureau (2018). Modeling of Holiday Effects and Seasonality in Daily Time Series. Center for Statistical Research and Methodology Report. <https://www.census.gov/srd/papers/pdf/RRS2018-01.pdf>

343 **Appendix A**

- 344 • The observation windows are not the same for all turnstile machines.
- 345 • Each machine keeping track of counts is identified by a unique subunit channel position (scp) within a
346 control area.
- 347 • Control area is the bank of turnstile machines associated with a subway exit(s), unique within a remote
348 unit, or a single station.
- 349 • Once a unit gets to the most recent date, the following turnstile unit would take the previous turnstile
350 unit as a starting point, leading to extremely inaccurate entry counts in the first period. To solve this,
351 we flag the first instance of a unit and convert it to NaN.
- 352 • Once a turnstile reaches a certain amount, they roll back to zero, which causes huge entry counts for
353 the period due to some turnstile logs getting to over nine digits. We eliminate those outliers, which
354 reduces the data by 0.7% but it reduces our counts by billions and saves time.
- 355 • Some of the turnstile machines are counting down, not up. Thus we take the absolute difference.

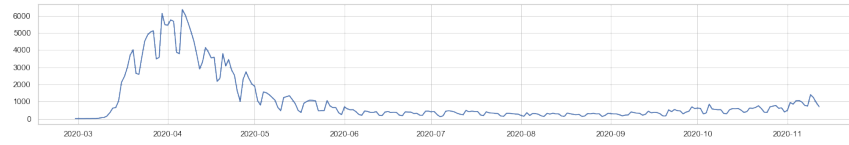


Figure 1: Daily COVID-19 case counts

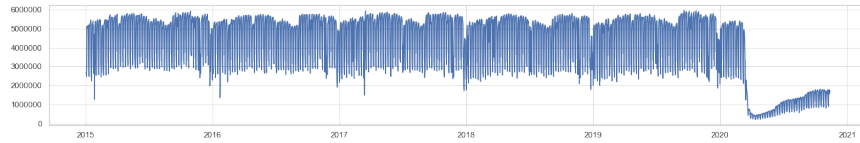


Figure 2: Daily subway turnstile entry counts

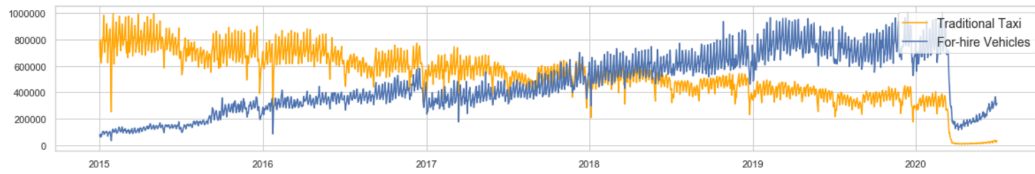


Figure 3: Daily taxi and FHV pick-up counts

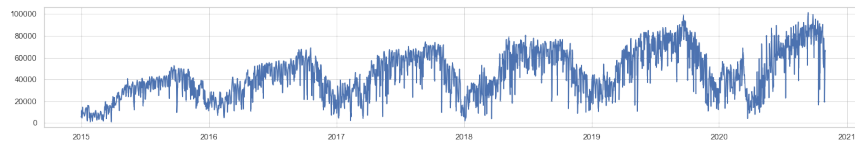


Figure 4: Daily Citi Bike ridership

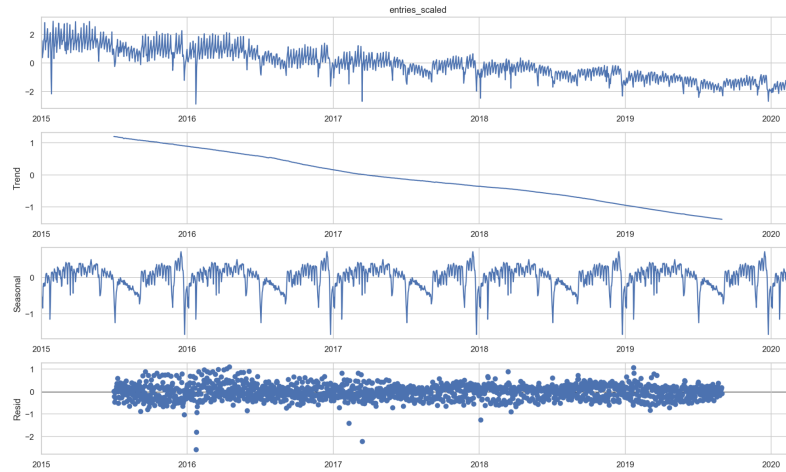


Figure 5: Seasonal decomposition of taxi data (ARMA)

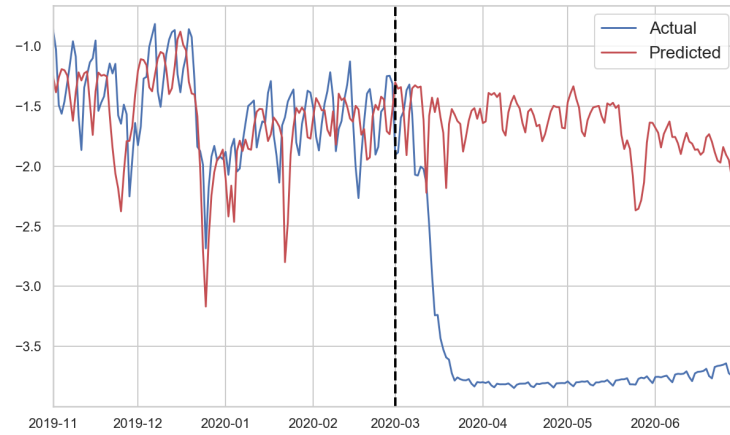


Figure 6: Predicted taxi pick-up counts vs. ground truth (ARMA)

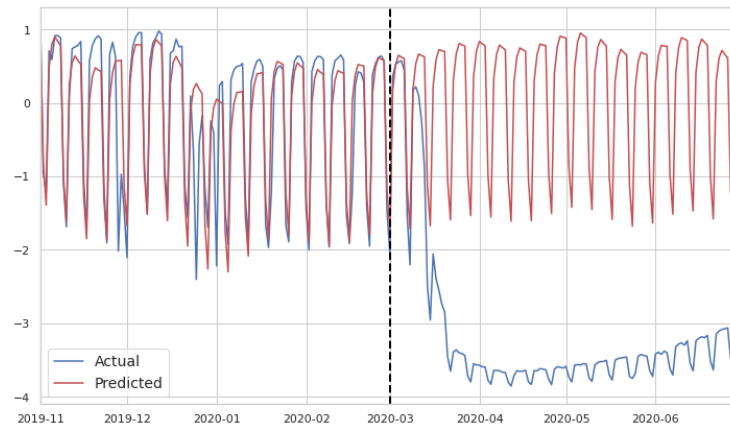


Figure 7: Predicted subway entry counts vs. ground truth (Prophet)

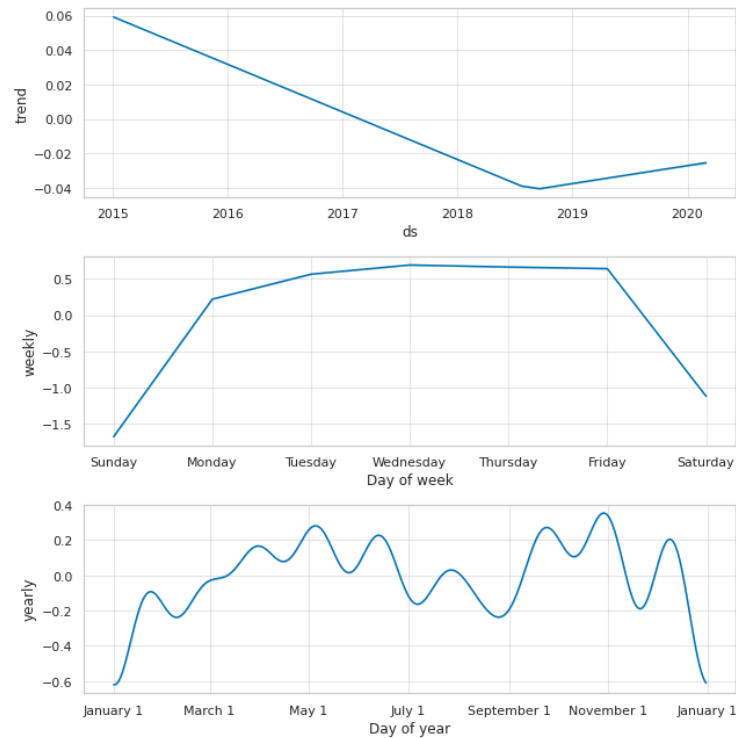


Figure 8: Non-linear trend and seasonality in subway data (Prophet)

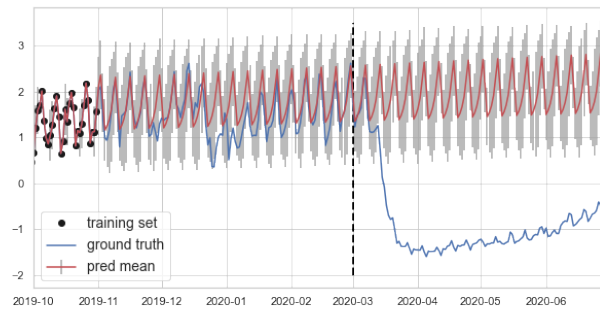


Figure 9: Predicted FHV pick-up counts vs. ground truth (GP)

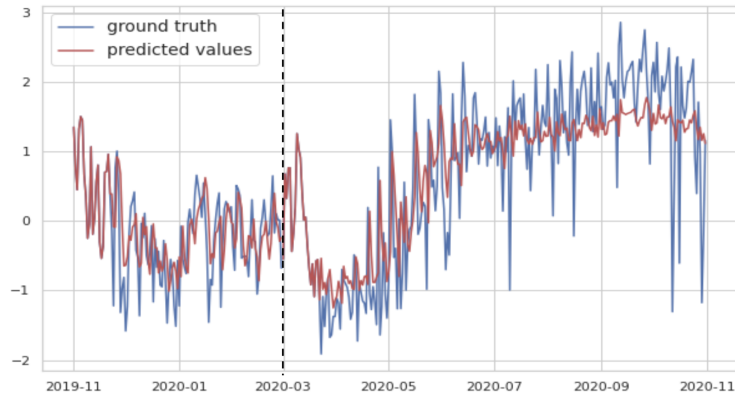


Figure 10: Predicted Citi Bike counts vs. ground truth (RNN PW)

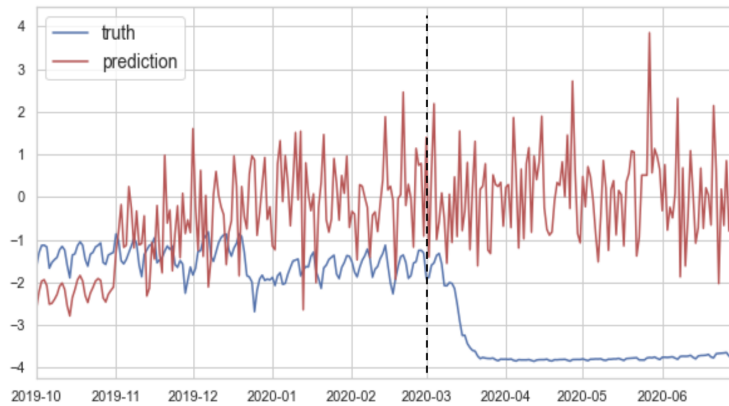


Figure 11: Predicted taxi pick-up counts vs. ground truth (KF)

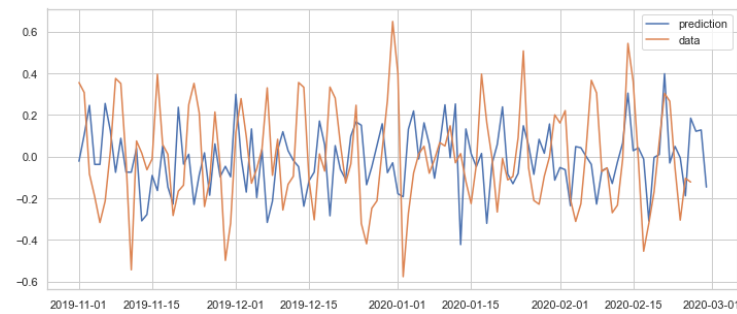


Figure 12: Predicted taxi pick-up counts vs. ground truth (KF) without seasonality