Forecasting Methods of Daily and Hourly Citi Bike Trips During the Covid-19 Pandemic

Submitted by
Miao Chenxin
U2101169

School of Electronic Information
Northwestern Polytechnical University

Supervisor：Dr. Zhang Jianwen

The final year project work was carried out under the 3+1+1 Educational Framework at the National University of Singapore (Suzhou) Research Institute

May 2022

ABSTRACT

The COVID-19 pandemic is having an impact on people's transportation. Citi Bike, as one of the important travel tools for New Yorkers, is a representative of flexible transportation tools. It is worth studying whether the travel level of Citi Bike has been affected to some extent due to the emergence of the epidemic. This paper analyzes the characteristic factors that affect the trips of Citi Bike, and uses four different time series forecasting methods to analyze the trips of Citi Bike during the COVID-19 period, and to predict the trips of Citi Bike from 2020.8.15 to 2020.8.31. Time series prediction methods include ARIMA, SARIMA, XGBoost and LSTM using Grid search. By comparing the prediction results with MSE, MAE and R2 indexes, XGBoost model has the best prediction effect, and it also indicates that epidemic factors have a certain effect on Citi bike trips.

ACKNOWLEDGMENTS

First of all, I would like to sincerely appreciate my supervisor Dr. Zhang Jian Wen of National University of Singapore (NUS) for his patient guidance, suggestions on topic selection and the gentle encouragement. Thanks to his interesting topics, I have expanded my horizon and become familiar with data science. Secondly, I would like to thank my examiner Assistant Prof. Guo Yongxin who gave me a lot of advice during CA1. Thirdly, I would like thank to National University of Singapore (Suzhou) Research Institute and Northwestern Polytechnical University for providing me the opportunity to spend a wonderful semester in Suzhou. Finally, I would like to thank my family and friends, as well as my lovely cats for their support.

CONTENTS

… … …

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The government responded positively to the outbreak of COVID-19 in early March 2020 with a rapid lockdown, which had a significant impact on people's transportation. Citi Bike, as one of the most important travel tools for New Yorkers, is both convenient and environmentally friendly and is a representative of flexible transportation tools for New Yorkers. During the spread of the virus, it is safer to travel by bicycle than by using public transport (subways, buses, taxis) with more people. Compared with private transportation (e.g. private cars), it is more economical. In the wake of the outbreak, travel was restricted due to a government stay-at-home order. With the continuous unsealing of New York City and the gradual opening of economic and social activities, it is worth studying whether the travel level of Citi Bike, as an alternative means of transportation, is affected to some extent.

In previous studies, features like demographic factors, weather factors, bike membership policies, and other transportation means are usually associated with bike trips before the COVID-19 outbreak. During the pandemic period, researchers pay more attention to policymaking, including the lockdown and reopen policy, due to government stay-at-home orders. The factors of the epidemic in previous studies are ignored, and the comparison between multivariate prediction and univariate prediction needs to be added. In this paper, given data during the pandemic period, Covid-19 features such as the number of cases, hospital visits, and deaths are added to better predict the city-level bike trips by direct comparisons with univariate prediction.

This paper focuses on the transportation trend during the COVID-19 pandemic, which contains the period from 2020.3 to 2020.8. Time series prediction methods include ARIMA, SARIMA, XGBoost, and LSTM using Grid search. By observing and comparing the number of bike trips changing over time, the trend caused by the pandemic could be estimated. Since it is difficult to transition hourly data into a stationary sequence for ARIMA and SARIMA models, daily data are used for modeling and as the control group, while hourly data are used for the latter two groups. Among those, XGBoost performs best on all sides of the evaluation index. The project framework is shown in figure 1.

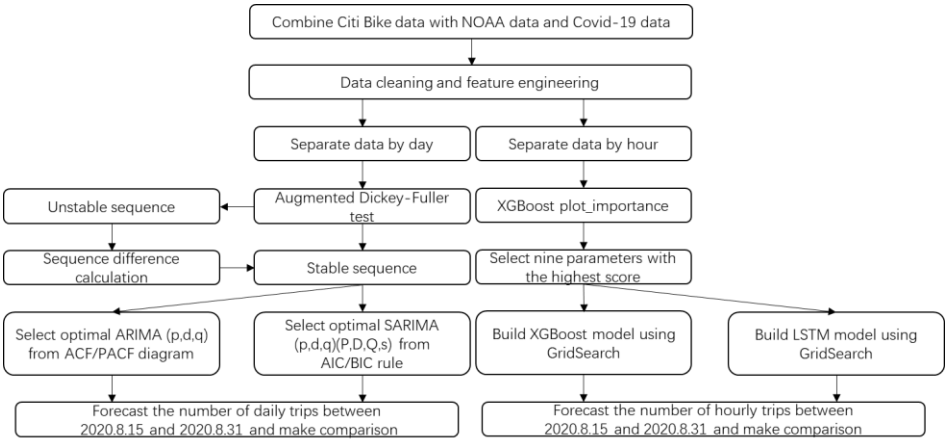

Figure 1 The project framework

2 Literature Review

As one of the most common means of transportation for citizens, bikes are both convenient and environmentally friendly, and there are a large number of bicycles available for rent in medium and large cities [1]. Before the COVID-19 outbreak, there were many literatures on analyzing and predicting bicycle travel patterns. When analyzing the characteristics of these literatures, demographic factors, weather factors, bike membership policies, while the selected weather factor may be the most important among those [2].Other transport travel factors are also used to predict bike travel patterns (taxis,subway.etc) [3]. The pattern that need to be predicted are mainly the number of bike trips and travel time [4]. The research scope is from single station level to city level [5,6]. In the wake of the coronavirus outbreak, public transport has been largely restricted due to government stay-at-home orders. During the spread of the virus, it is safer to travel by bike than by using public transport with more people (e.g. subways, buses, taxis), thus increases the trip duration of bike and decreases number of bike trips in certain period of time[7]. In addition, the influence of policy is widely studied, especially the lockdown and reopen policy, showing the great impact on citi bike sharing system[8].

The methods used in previous study are variable. For single station level, Long short term memory , Gradient Boosting , Gated Recurrent Units are commonly used[9].In city level, Linear Regression, Support Vector Machine, Boosted Trees and other models are commonly used[10]. In several stations level, K Nearest Neighbors, Random Forest , Naïve Bayes are commonly used[11]. The clustering algorithms are more used when there are more stations to be predicted. The study period is at least three months and at most one year, showing great flexibility.

3 Data preparation

3.1 Dataset description

3.1.1 NYC citi bike dataset

Citi Bike is the largest bike-sharing program in the United States, with 25,000 bikes and more than 1,500 stations in Manhattan, Brooklyn, Queens, the Bronx, Jersey City and Hoboken[12]. Citi Bike data is allowed to know the departure point and destination, departure time and distance. This data has deleted trips made by employees working for the system and any trips with travel times less than 60 seconds. Some data of Citi Bike are shown in Table 1.

Table 1 Citi Bike sharing data in New York City

Feature	Description
trip_duration	Travel time in seconds
start_time	Start time in seconds
stop_time	Stop time in seconds
start_sta_id	The id of start station
start_sta_name	The name of start station
start_sta_lat	The latitude of start station
start_sta_lon	The longitude of start station
end_sta_id	The id of end station
end_sta_name	The name of end station
end_sta_lat	The latitude of end station
end_sta_lon	The longitude of end station
bike_id	The id of bike
user	Customer = 24-hour pass or 3-day pass user; Subscriber = Annual Member
birth	Year of birth
gender	Zero=unknown; 1=male; 2=female

3.1.2 NOAA dataset

NOAA data with metadata records is searchable and downloadable at the NATIONAL Oceanic and Atmospheric Administration[13]. It is an interoperable tool designed to store data from different scientific disciplines, formats, time periods and locations. Select the locations where you want to view the weather from the world map and filter through your browser to get a variety of downloadable data sets for the region. The information provided at present is largely limited to weather and climate information. Some NOAA data are shown in Table 2.

Table 2 Partial NOAA data

Feature	Description
DATE	Date of obserwacje meteorologiczne

temp	average temperature
slp	Sea level pressure
stp	Average station pressure of the day
dewp	The average dew point
visib	visibility
wdsp	average wind velocity

3.1.3 Covid-19 cases dataset

COVID-19 data shows how COVID-19 has affected the people of New York City since the city's first confirmed case was diagnosed on February 29, 2020. The data includes cumulative confirmed cases, hospitalizations, and death rates for New York City as a whole, as well as cumulative confirmed cases, hospitalizations, and death rates for several densely populated areas, and is updated in real time[14]. Some Covid-19 data are shown in Table 3.

Table 3 Partial Covid-19 data

Feature	Description
DATE_OF_INTEREST	Date of COVID-19 diagnosis (i.e., date of specimen collection), hospital admission, or death
CASE_COUNT	Count of patients tested who were confirmed to be COVID-19 cases on date_of_interest
probable_case_count	Count of probable cases occurring on date_of_interest. A person is classified as a probable COVID-19 case if they meet any of the following criteria with no positive molecular test on record: (a) test positive with an antigen test, (b) have symptoms and an exposure to a confirmed COVID-19 case, or (c) died and their cause of death is listed as COVID-19 or similar
HOSPITALIZED_COUNT	Count of COVID-19 patients who were hospitalized on date_of_interest
DEATH_COUNT	Count of deaths occurring among confirmed COVID-19 cases on date_of_interest
DEATH_COUNT_PROBABLE	Count of probable deaths occurring on date_of_interest. Probable deaths include those with cause of death reported as "COVID-19" or equivalent, but there is no positive laboratory test for COVID-19.
CASE_COUNT	7-day average of count of patients tested who were confirmed to be

_7DAY _AVG	COVID-19 cases. The seven days include the current day and the previous six days.

3.2 Data preprocessing

The New York City stay-at-home order was in effect on March 22 when the coronavirus outbreak began in early March 2020. Between June 8 and July 22, there were four phases of the city's gradual opening[15]. The selected time range of this data set is from the beginning of the outbreak to nearly three weeks after the fourth opening as the training set, and the time length is from March 1, 2020 to July 14, 2020. The next two weeks or so will be the test set, from July 15 to July 31. In the selection of time and frequency, the data set is divided into day-level data and hour-level data, which are respectively used for the prediction of different models to maximize the prediction accuracy of models.

For the three data sets, missing values or outliers are first checked, and then cleaned and spliced. To obtain the useful features, part of features are through feature engineering. For Citi Bike data, the category and gender of users are one-hot coded. The number of bicycle trips are calculated by hour and by day; For time data, they are separated out the number of days and hours of bicycle departure time to judge the day of a month, the day of a week, and whether it is in the peak hours. Since time data are periodic variables, their sine and cosine values need to be calculated separately. For weather data, several features with small standard deviation of their own are removed; For the epidemic data, remove all probable statistical features, because there is a large overlap with the actual numbers; The number of trips in each time period (num_trips) is the feature to be predicted, as shown in Table 4.

Table 4 Cleaned data features

Category	Feature
Citi bike data	start_hour, num_trips, start_station_id, avg_trip_duration, gender_0, gender_1, gender_2,user_customer, user_subscriber
Time data	day_of_week,day_of_month,month,is_weekend,hour,hour_sin,hour_cos,day_of_week_sin,day_of_week_cos,day_of_month_sin, day_of_month_cos,month_sin,month_cos,is_rushday,is_rushnight
Weather data	temperature,dewpoint,sl_pressure,wind_speed,max_wind_speed,max_temp,min_temp
Covid-19 data	case_count,hosp_count,death_count,bx_case_count, bx_hosp_count,bx_death_count,bk_case_count,bk_hosp_count,bk_death_count,mn_case_count,mn_hosp_count,mn_death_count,qn_case_count,qn_hosp_

	count,qn_death_count,si_case_count,si_hosp_count , si_death_count
Predicted data	num_trips

4 ARIMA and SARIMA

4.1 Model Theory

A time series is a set of timestamped data entries, which is a sequence obtained by arranging the values of a certain statistical index in chronological order [16]. Time series prediction is a method of regression prediction by compiling and analyzing time series. Through the statistical analysis of the past time series, the development process and direction of data are obtained, and the analogy or extension is carried out to get the trend of data development over time, predicting the level that may be reached in the next period of time. Statistical models and machine learning models can be used to predict time series. Common statistical models for time series prediction include Autoregressive Model, Moving Averages Model, Autoregressive Integrated Moving Average Model and Seasonal Autoregressive Integrated Moving Average Model[17].

4.1.1 Autoregressive Model (AR)

Autoregressive model is a statistical time series model that uses the past value of a variable to predict the present value. A p-order autoregressive process, AR(p), takes the form:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \epsilon_t \quad (1)$$

Where ε_t is white noise, c is constant, φ_i is coefficient of past values, and y_t is the predictive value at time t, which can be seen as a multiple regression but with lagged values[18].

4.1.2 Moving Averages Model (MA)

The moving average model uses past forecast errors in a regression-like model rather than using the past value of the prediction variable in a regression. A q-order Moving Averages process, MA(q), takes the form:

$$y_t = c + \sum_{i=1}^{q} \theta_i y_{t-i} + \epsilon_t \quad (2)$$

Where ε_t is white noise, c is constant, θ_i is coefficient of past values. y_t is the predictive value at time t, which can be seen as the weighted moving average of past errors in several forecasts [18].

4.1.3 Autoregressive Integrated Moving Average Model (ARIMA)
Autoregressive moving average model is an important method to study time series. It is composed of autoregressive model (AR) and moving average model (MA). This method is a typical method to study rational spectrum of stationary random process and is suitable for a large class of practical problems. A p,q-order Autoregressive Integrated Moving Average Model, ARMA(p,q), takes the form:

$$y_t = c + \sum_{i=1}^{q} \theta_i y_{t-i} + \sum_{i=1}^{p} \varphi_i y_{t-i} + \epsilon_t \quad (3)$$

Where ε_t is white noise, c is constant, φ_i, θ_i are coefficient of AR and MA models. y_t is the predictive value at time t, which can be seen as the sum of the lagged past values and the weighted moving average of past errors in several forecasts.

ARIMA is the combination of difference operation and ARMA model. In the parameter spectrum estimation of ARIMA (p, d, q), AR parameters are estimated first, and then MA parameters are estimated on the basis of these AR parameters, so as to obtain the spectral estimation of ARMA parameters. Where, AR represents the autoregressive model, MA represents the moving average model, I represents difference, p represents the order of autoregression, d represents the times of difference processing for the non-stationary time series with long-term trend, and q represents the order of the moving average. Compared with AR model and MA model, it has more accurate spectral estimation and a better spectral resolution performance[19].

4.1.4 Seasonal Autoregressive Integrated Moving Average (SARIMA)
Seasonal Autoregressive Integrated Moving Average Model(SARIMA) is a seasonal ARIMA model with exogenous regressive models. On the basis of ARIMA, the seasonal part is added. Seasonality is a repeating pattern in an index data with a fixed frequency: daily, every two weeks, every three months, etc. Seasonal components may capture long-term patterns, while non-seasonal components adjust predictions of short-term changes.

SARIMA model can be expressed as SARIMA (p, d, q) x (P, D, Q) s, which satisfies the multiplication principle. The first part represents the non-seasonal part, the second part represents the seasonal part, and S represents the seasonal frequency[20].

4.2 Steps
(1) Acquisition of time series
Travel statistics for 2020.3-2020.8 are obtained through information on Citi Bike's official website. The data are first checked for missing values or outliers,

and then time series data on the number of Citi bike trips per day and per hour are obtained.

(2) Stationarity test of time series

The time series that can enable ARIMA and SARIMA model to analyze and predict should be stationary series. The stationarity of data is tested by rolling statistics and ADF test.

(3) Time series stabilization

If the time series does not meet the requirements of stationarity, the non-stationary time series data needs to be stabilized until the values of the processed autocorrelation function and partial autocorrelation function are non-significant and non-zero. For non-stationary time series, if there is a trend of growth or decline, it is necessary to carry out differential processing and then carry out stationarity test until it is stable. Time series are generally analyzed by comparison of first-order, second-order and third-order differences.

(4) Parameter estimation

ACF(automatic correlation function) and PACF(partial automatic correlation function) diagrams are used for model ordering to find the (p,d,q) value of ARIMA. And the AIC/BIC rules are used for model ordering to find the (p,d,q)(P,D,Q)s value of ARIMA

(5) Model prediction

The time series prediction is carried out using the tested model and the evaluation is calculated.

4.3 Data visualization and preparation

When using time series models, such as ARIMA and SARIMA, time series are required to be stable. A time-stationary sequence is defined as a series whose mean is constant, it has constant variance or standard deviation, and its self-covariance should not depend on time.

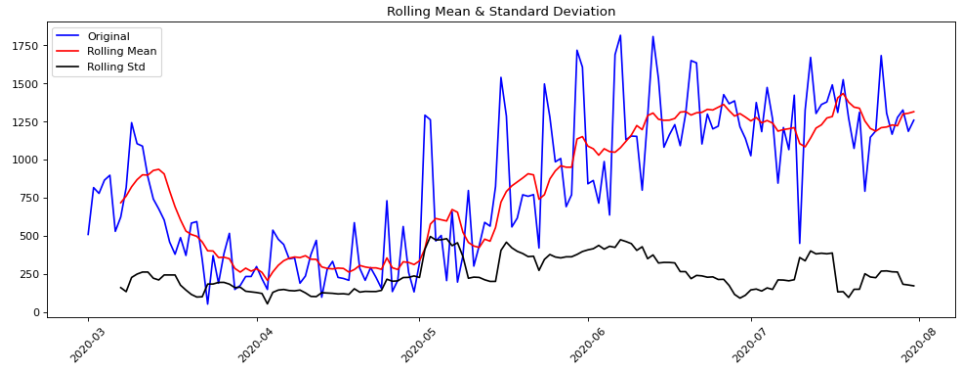

Figure 2 A rolling count of the number of daily Citi Bike trips

By drawing a rolling statistical graph of bike trips, the change trend over time can be observed, as shown in figure 2. The two rolling statistics: rolling mean and rolling standard deviation need to remain constant over time, so their curves must both be parallel to the X-axis. Figure 2 indicates that the daily data is not stationary. To further enhance the hypothesis that time series is not stable, it is

necessary to further adopt strict statistical test methods, such as Augmented Dickey-Fuller test, and use adFuller function for illustration. Augmented Dickey-Fuller test is also called unit root test. Unit root test refers to test whether there is unit root in the sequence. The sequence with unit root is an unstable sequence, which will lead to pseudo regression in regression analysis. The H0 hypothesis of ADF test is the existence of unit root. If the significance test statistics obtained are less than three confidence degrees (10%, 5%, 1%), the null hypothesis is rejected with confidence of due (90%, 95%, 99%)[21]. The ADF test result is shown as table 5.

Table 5 Augmented Dickey-Fuller test result

	Original daily data
Test Statistic	-0.165543
p-value	0.942506
Lags Used	13.000000
Number of Observations Used	170.000000
Critical Value (1%)	-3.469413
Critical Value (5%)	-2.878696
Critical Value (10%)	-2.575917

Per Table 5, the p-value is about 0.94,which is close to 1. The test statistic value is larger than Critical values. P values are required to be less than the given significance level, preferably less than 0.05. The Critical values of the 1%, 5%, and 10% confidence intervals should be as close as possible to the Test Statistic. If the test statistics are less than 1%, 5%, and 10% at the same time, this hypothesis is well rejected. Per Table 5, there is no similarity between Critical Value and Test Statistic. Therefore, the original time series is unstable at the moment.

4.4 Time series stabilization

There are many methods for time series stabilization, including difference calculation, moving average method, exponential smoothing method, etc[22]. It can be seen from Table 6 that the p-value of the first order difference method is the smallest, which is close to 0. And the test statistic is all less than three critical values, indicating that the model rejects this hypothesis very well. Therefore, the first order difference method is selected to stabilize the data.

Table 6 Augmented Dickey-Fuller test result of stabilized series

	First order difference Method	Moving average method	Exponential attenuation method
Test Statistic	-9.213486e+00	-8.501040e+00	-2.348224
p-value	1.860514e-15	1.238129e-13	0.156886
Lags Used	4.000000e+00	1.000000e+00	6.000000
Number of	1.470000e+02	1.710000e+02	177.000000

Observations Used			
Critical Value (1%)	-3.475637e+00	-3.469181e+00	-3.467845
Critical Value (5%)	-2.881410e+00	-2.878595e+00	-2.878012
Critical Value (10%)	-2.577365e+00	-2.575863e+00	-2.575551

4.5 Parameter estimation and order selection

The autocorrelation function (ACF) is a statistical technique that can be used to determine the correlation of values in a time series with each other. ACF plots the correlation coefficient with a lag, which is a visual representation of autocorrelation. The correlation coefficients can range from -1(a perfect negative relationship) to + 1(a perfect positive relationship), with a coefficient of 0 implying that the variables are not related.

The partial auto-correlation function(PACF) is a partial autocorrelation function that can be used to find a correlation with the next lag residuals (these are values that remain after removing other influences). If there is any residual information that can be modeled through the next lag, a good correlation can be got and it will retain this lag as a feature when modeling[23].

As shown in Figure 3, the ACF and PACF diagram below: the blue bars on the diagram are error bands, and anything within these bars is not statistically significant. This means that the correlated values outside this region are likely to be correlated, rather than a statistical fluke. By default, the trust interval is set to 95%.

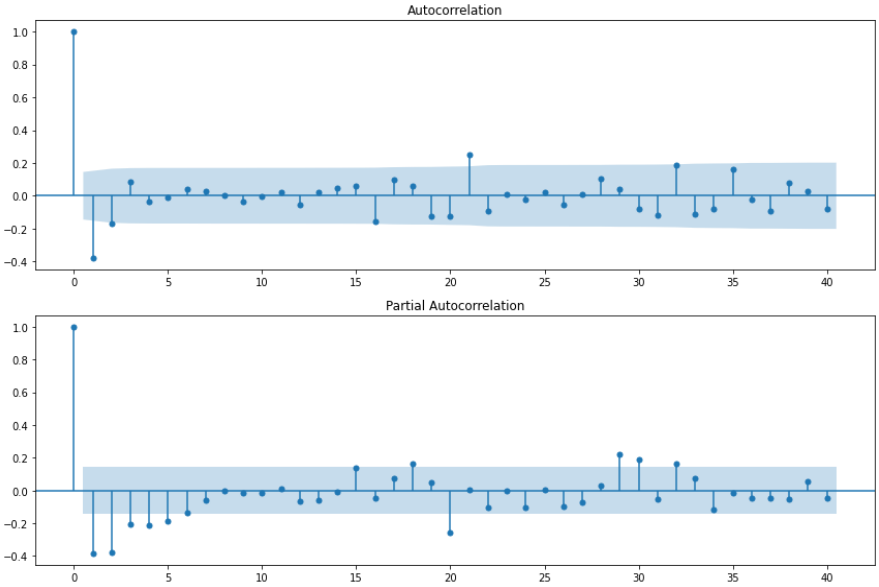

Figure 3 ACF/PACF diagram

The value of q can be roughly judged from the maximum lag point of the autocorrelation coefficient (ACF) graph. The p value can be roughly judged from the maximum lag point of the partial autocorrelation coefficient (PACF) graph[24]. Per figure 3, the combination of (p,d,q) is (1,1,5), (1,1,6), (2,1,5), (2,1,6). The model used here is ARIMA model (1,1,6) by comparison. The ARIMA prediction formula of the number of daily bicycle trips of the first order difference is as follow:

$$\hat{Y}_t = \mu + \hat{Y}_{t-1} + \phi_1(Y_{t-1} - Y_{t-2}) - \theta_1(Y_{t-1} - \hat{Y}_{t-1}) - \theta_2(Y_{t-2} - \hat{Y}_{t-2}) - \theta_3(Y_{t-3} - \hat{Y}_{t-3}) - \theta_4(Y_{t-4} - \hat{Y}_{t-4}) - \theta_5(Y_{t-5} - \hat{Y}_{t-5}) - \theta_6(Y_{t-6} - \hat{Y}_{t-6}) \quad (4)$$

Where μ is constant, φ_i, θ_i are coefficient of AR and MA models. $\hat{Y}_t$ is the predictive value at time t, Y_t is the true value at time t, $\widehat{y_t} = Y_t - Y_{t-1}$ due to the first order difference.

For SARIMA, seasonal factors need to be removed before making a first-order difference. Compared with ARIMA, SARIMA adds four parameters (P,D,Q)s. Where, P is the seasonal autoregressive order, D is the seasonal difference order, Q is the seasonal moving average instruction, and S is the number of time steps in a single season[25]. By comparison, the seasonal length of the model is 14, that is, the value of S in the model is 14, and it is known that d=1 is selected for the first-order difference. As SARIMA's order is difficult to judge by ACF/PACF diagram, Akaike Information Criterion is selected. AIC judgment is a standard to measure the excellence of statistical model fitting. Based on the concept of entropy, it can balance the complexity of the estimated model with the excellence of the model fitting data. The order of SARIMA can be determined by searching for (p,q, P,Q).

As shown in table 7, the selected model has p of 1, q of 2,P of 2 and Q of 1, which has the lowest AIC of 2149.51636. Therefore, the model can be described as SARIMAX(1, 1, 2)x(2, 1, [1], 14).

Table 7 Selected parameters of SARIMA

Hyperparameter	Description	Value Tested	Value Selected
p	Trend autoregression order	1,2,3,4,5	1
q	Trend moving average order.	1,2,3,4	2
P	Seasonal autoregressive order	1,2	2
Q	Seasonal moving average order	1,2	1

5 XGboost and LSTM

5.1 GirdSearch

Grid search is a parameter tuning method in machine learning. It lists all possible algorithm parameter values into a grid. By permutation and combination of various parameter values, the model will automatically select the optimal parameter combination. The scoring function introduced by this automatic parameter adjustment mechanism will score each parameter combination, and finally select the parameter with the highest score for modeling to improve the accuracy of prediction[26]. The k-fold cross-validation method divides all the original training sets into K groups, and makes each subset do a validation set, and the rest K-1 subsets serve as training sets. The average classification accuracy of the final validation set of the K models is used as the performance index of the classifier under the K-CV. In general, the fold k is usually 3,5,10,15, etc.

5.2 Steps

The grid search method is an exhaustive search method for specifying parameter values. The optimal learning algorithm is obtained by optimizing the parameters of the estimated function through cross validation. Permutations and combinations of possible values for each parameter and lists all combinations to generate a "grid."

Each combination parameter is then used for model training and evaluated using cross validation. In order to select the best parameter among the values of manually set adjusted variables, an ideal scoring method is needed (scoring methods are generally accuracy, F1-score, F-beta, Pricise, recall, etc.). After the fitting function has tried all the parameter combinations, it returns an appropriate classifier and automatically adjusts to the best parameter combination. Before grid search, according to the idea of grid search algorithm, the parameter combination interval to be selected is set first. The model is continuously trained in the process of parameter optimization, and each function is evaluated by evaluation function. Finally, the results obtained by parameter combination are evaluated and the optimal parameter combination is obtained. By substituting the optimal parameter combination into Xgboost model and LSTM model, the specific steps are as follows:

(1) Obtain the hourly data and then standardize; Establish the model scoring function.
(2) Divide the data set into training set and test set for training models, and select the partition of the minimum RMSE.
(3) Select the k value that minimizes RMSE in k-fold verification
(4) Train the XGBoost model and calculate the importance of features. The 9 features with higher feature importance F score are selected to re-establish the dataset, thus establishing a less complex model and accepting a moderate

reduction in the estimation accuracy.
(5) List the permutations and combinations of all XGBoost and LSTM hyperparameters to form a grid, and train the models again. Based on all parameter combinations, the scoring function is used to score each model, and the XGBoost and LSTM models with the best scoring effect are finally selected.
(6) Compare the predicted value with true value.

5.3 eXtreme gradient boosting

eXtreme gradient boosting was first proposed by Tianqi Chen[27]. It is an optimized distributed gradient boosting method designed to be highly efficient and portable. It implements machine learning algorithms under the Gradient Boosting framework. Boosting algorithm is called Gradient boosting if the generation of weak classifiers in each step is based on the Gradient direction of loss function. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. XGBoost algorithm adopts a stepwise forward additive model. It is an additive expression composed of K base models:

$$\widehat{y_i} = \sum_{i=1}^{k} f_t(x_i) \quad (5)$$

Where $f_t(x_i)$ is the i-th basis model, $\widehat{y_i}$ is the predicted value of the i-th sample. The loss function can be expressed by predicted value $\widehat{y_i}$ and true value y_i:

$$L = \sum_{i=1}^{n} l(y_i, \widehat{y_i}) \quad (6)$$

Where n is the number of samples. XGBoost uses Taylor's second-order expansion to optimize the objective function, and adds regularization terms to the loss function to control the complexity of the model. The objective function of its optimization is as follows:

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i{}^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + C \quad (7)$$

Where $Obj^{(t)}$represents the objective function of the t training; y_i is the true value of the ith sample; $\hat{y}_i{}^{(t-1)}$ is the predicted values of the t-1 round model; L is the loss function; $f_t(x_i)$ represents the function value of round t when input is x_i; $\Omega(f_t)$ is the regularization term. C is a constant. Since $\hat{y}_i{}^{(t-1)}$ is a known value at the t step, $l(y_i, \hat{y}_i{}^{(t-1)})$ is a constant, which has no influence on the optimization of the function. Therefore, the objective function can be further written as:

$$Obj^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \sum_{i=1}^{t} \Omega(f_i) + C \quad (8)$$

Where g_i is the first derivative of the loss function L, h_i is the second derivative

of the loss function L.

$$g_i = \frac{\partial L(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \quad , \quad h_i = \frac{\partial^2 L(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} \quad (9)$$

While training the model, the decision tree traverses. w represents the weight of the sample, and w_j represents the predicted value of j th sample. I_j represents the sample set of the j-th leaf node in the decision tree. Therefore, formular (8) can be simplified as:

$$Obj^{(t)} = \sum_{j=1}^{t} \left[G_j w_j + \frac{1}{2} {w_i}^2 (H_j + \lambda) \right] + \gamma T = -\frac{1}{2} \sum_{j=1}^{T} \frac{{G_j}^2}{H_j + \lambda} + \gamma T \quad (10)$$

Where:

$$G_j = \sum_{i \in I_j} g_i \, , \quad H_j = \sum_{i \in I_j} h_i \quad (11)$$

Formular (10) is the scoring function measuring of the quality of decision tree. It is used to select the best segmentation point to build a CART tree and estimate of feature importance from a trained model[27].

5.3.1 Feature importance

Feature selection is an important step in machine learning to screen out significant features and discard non-significant features. Feature selection can improve training speed, reduce over-fitting risk and improve model effect.
The XGBoost algorithm uses gradient enhancement, and it is relatively easy to retrieve the importance score for each attribute after building the enhanced tree[28]. Importance provides a score that indicates the usefulness or value of each feature in building an enhanced decision tree in the model. The more attributes are used for key decisions used in decision trees, the higher their relative importance. This importance is explicitly calculated for each attribute in the data set so that the attributes can be ranked and compared with each other. The importance of individual decision trees is calculated by the number of performance indicators improved by each attribute split point and weighted by the number of observations the node is responsible for. The performance measure can be the purity used to select the resolution point (gini coefficient) or some other, more specific error function. Feature importance is averaged across all decision trees in the model.

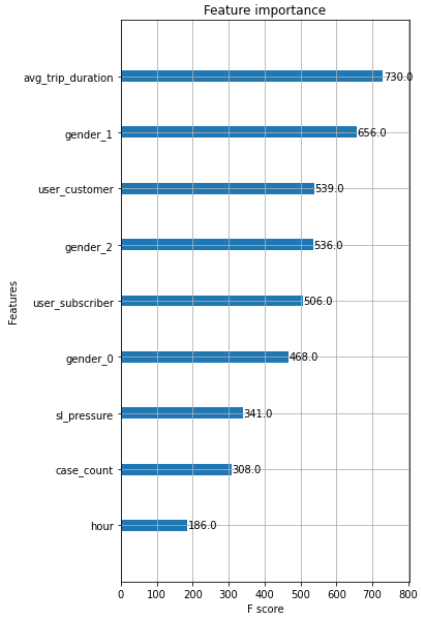

Figure 4 The importance of selected features

As shown in Figure 4, 9 features with the highest F score of feature importance are selected as features. The selected features mainly include: avg_trip_duration,gender_1,user_customer,gender_2,user_subscriber,gender_0,sl_pressure,case_count and hour. Where, avg_trip_duration is the average cycling duration; gender_0, gender_1, and gender_2 are heat-coded respectively, representing the gender of cyclist. User_customer, user_subscriber are heat-coded respectively, representing the categories of cyclist. Sl_pressure is a weather feature, representing the New York's sea level pressure; Case_count is an outbreak characteristic, representing the number of confirmed Covid-19 cases.

5.3.2 Model training of XGBoost

The input data is 4161, and the features used to predict the nine highest-scoring features are selected by Xgboost's plot_importance, the input data is divided into training sets and test sets in a ratio of about 9:1.

Hyperparameters are parameter values set prior to the start of the learning process, not parameter data obtained through training. It is usually necessary to optimize the hyperparameters and select a group of optimal hyperparameters for the learning machine to improve the performance and effect of learning [29]. During the training of XGBoost, the hyperparameters are set and its performance is evaluated by k-fold cross validation. In cross validation, the training data set is divided into K groups. One group is used as the test data set, the other as the training data set, and finally the average test error is used as the generalization error. In general, training data sets are fitted and evaluated on test data sets; This process was repeated k times, using each of the K groups as the test data set. All samples in the training set will inevitably become training data, and pages will have the opportunity to become test sets, so the training set data can be better utilized. In the process of hyperparameter selection, k value is selected as 12,

because it reaches the lowest point of RMSE, which is 1.589.

Before building the final XGBoost model, three types of parameters need to be set: general parameters, booster parameters, and task parameters. Among them, the general parameters are related to the booster we use, generally include tree or linear model; The booster parameters depend on which booster we choose; Learning task parameters determine learning scenarios[30].

The selected hyperparameters of the XGBoost model are shown in Table 8. It includes colsample_bytree, learning_rate, max_depth, min_child_weight, n_estimators, and subsample [25]. max_depth and min_child_weight are obtained together, the selected parameters are 3 and 5, reaching the lowest RMSE of 1.785. N_estimators interacts with the learning_rate, and for training with them, the selected parameters are 500 and 0.3, reaching the lowest RMSE of 1.344. To prevent overfitting, subsample and colsample_bytree are trained together, and the selected parameters are 0.8 and 0.5, reaching the lowest RMSE of 1.274. The final model (max_depth of 3,min_child_weight of 5, n_estimators of 500, learning_rate of 0.3,subsample of 0.8,colsample_bytree of 0.5) is built. This model has the lowest RMSE on validation set, which is 1.274

Table 8 Selected hyperparameters of XGBoost

Hyperparameter	Description	Value Tested	Value Selected
colsample_bytree	Subsample ratio of columns when constructing each tree	0.5 0.6 0.7 0.8 0.9 1.0	0.5
learning_rate	Step size shrinkage used in each boosting step	0.001 0.01 0.1 0.3 0.5	0.3
max_depth	Maximum depth of each tree	1 3 6 9	3
min_child_weight	Minimum sum of instance weight required in each child	1 3 5 7	5
n_estimators	Number of gradient boosted trees	20 50 100 200 300 400 500	500
subsample	Subsample ratio of training instances	0.5 0.6 0.7 0.8 0.9 1.0	0.8

5.4 Long-short memory model

Long-Short Term Memory network (LSTM), first proposed in 1997, is an improved recurrent neural network, which can solve the problem that RNN cannot handle Long distance dependence[31]. In time series prediction, LSTM is suitable for dealing with important events with very long intervals and delays to better discover long-term dependencies. The internal structure of LSTM consists of a Cell and three Gates: Gates contain input Gate, output Gate and forgetting Gate. The Cell controls what to remember, what to forget and how to use the Gates to update the memory, thus solving the problem of gradient explosion or gradient disappearance[32].

5.4.1 Model Training of LSTM

To obtain the optimal model, the LSTM model searches for the optimal parameters through Grid search, and the hyperparameters to be determined include LSTM Layers, Dense Layers, units and memory hours. The evaluation index is MAPE (Mean Absolute Percentage Error). By traversing parameters, the MAPE value of each model can be obtained, and the optimal MAPE model can be obtained and saved. The input data is 4161, and the features used to predict the nine highest-scoring features selected by Xgboost's plot_importance, the input data is divided into training sets and test sets in a ratio of about 9:1.

First, the predicted hour is set to be 10, i.e., moving up 10 hours of data, the model predicts a length of 10 hours. Then the number of memory hours is selected due to the length of input data. The value of memory hours is selected between 5,10 and 15, and it is packaged into a memory module. After testing, the model with memory hours of 10 has the best effect.

Then, the layers of the neural network are tested, the number of layers to model is generally between 1 and 3. Therefore, one, two, and three lstm_layers are created separately and the MAPE of the model is calculated separately, with one lstm_layer working best. In general, the LSTM layer is added into dense, and the number of layers is generally between 1 and 3. Similarly, three models are tested respectively, and MAPE of the models is caculated. Among them, with two dense_layers working best. Activation function for lstm_layer and dense_layer is both RELU and the dropout rate is set to 0.1. Units are hidden neurons at the network layer. The value range of units is set to 16,32,64,128 according to the complexity of the model. By calculating MAPE, each layer has units of 128 [33].

As shown in table 9, the selected model has 10 memory hours(mem_hours), 128 units and 47 epoches with 1 lstm_layer and 2 dense_layers. During training, it has a MAPE of 10.13. Another model(44 epoches,5 men_hours,1 lstm_layer,2 dense_layer,128 units) has a MAPE of 10.18 with little difference in complexity and precision, so the first model is chosen.

Table 9 Selected hyperparameters of LSTM

Hyperparameter	Description	Value Tested	Value Selected
the_lstm_layers	The number of lstm layers	1,2,3	1
dense_layers	The number of dense layers	1,2,3	2
units	The number of units	16,32,64,128	128
mem_hours	Memory hours	5,10,15	10

6 Results and Comparison

6.1 Evaluation indicators

After building a model for prediction, its model performance needs to be evaluated[34]. There are three common indicators that can evaluate the performance of time series prediction models, namely RMSE (root mean Squared Error), which is used to measure the difference between observed value and true value. MAE (mean absolute error), which reflects the error generated by the predicted value, and R square value (R2), which can be used to compare the performance of models in different dimensions. Three indicators were selected to evaluate and compare the models.

RMSE is the root mean square error, which represents the sample standard deviation of the difference (residual) between the predicted value and the true value. The root mean square error is used to illustrate the dispersion of samples. For nonlinear fitting, A small RMSE value indicates that the model has a good prediction effect.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \quad (12)$$

MAE is the average of absolute errors. To avoid the offset of positive and negative errors, the absolute value can be used to better reflect the error between the real value and the predicted value. A small MAE value indicates that the model has a good prediction effect.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \quad (13)$$

R2 is used to represent the model fitting effect, and its value range is [0,1]. Generally, a large value indicates a good fitting effect and a small value indicates a poor fitting effect.

$$R2 = 1 - \frac{\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{m}(y_i - \bar{y})^2} \quad (14)$$

Where y_i indicates the actual value of the i-th sample, $\hat{y}_i$ indicates the predicted

value of the i-th sample, m indicates the total sample size, and $\hat{y}_i$ indicates the mean value.

6.2 Results

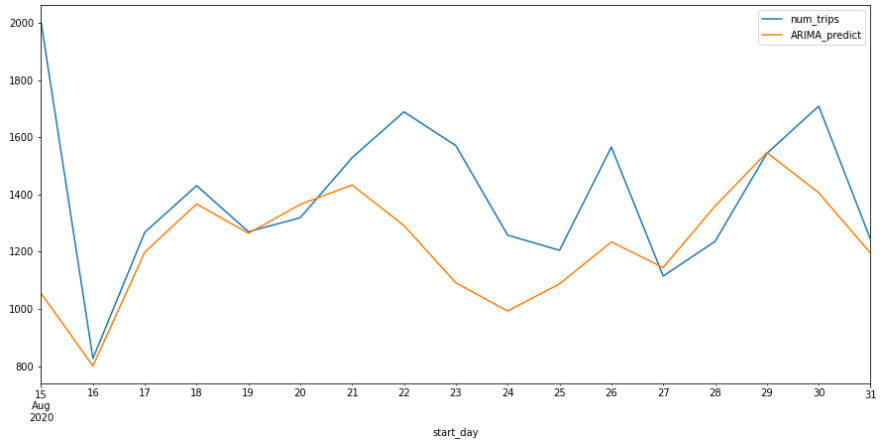

Figure 5 ARIMA prediction result

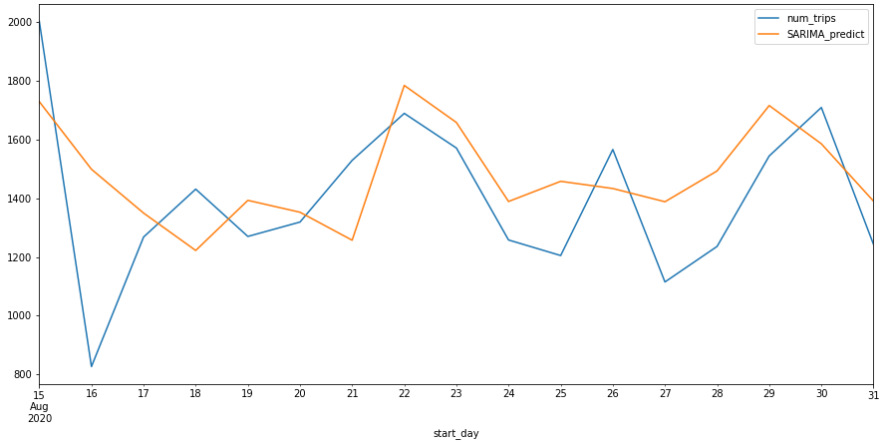

Figure 6 SARIMA prediction result

The prediction results of ARIMA and SARIMA are shown in figure 5 and 6. ARIMA and SARIMA models are used to make short-term prediction of the daily public bike trips in New York City. The orange line is the predicted daily number of bike trips, and the blue line is the true number of bike trips.

By comparison, the prediction of ARIMA model is slightly worse than that of SARIMA model, which may be because ARIMA's prediction of seasonal characteristics is poor, and the number of bicycle trips is affected by seasons to some extent. Therefore, the prediction results of ARIMA model are relatively poor and its reliability is low. Both models show a large error and a certain degree of time lag.

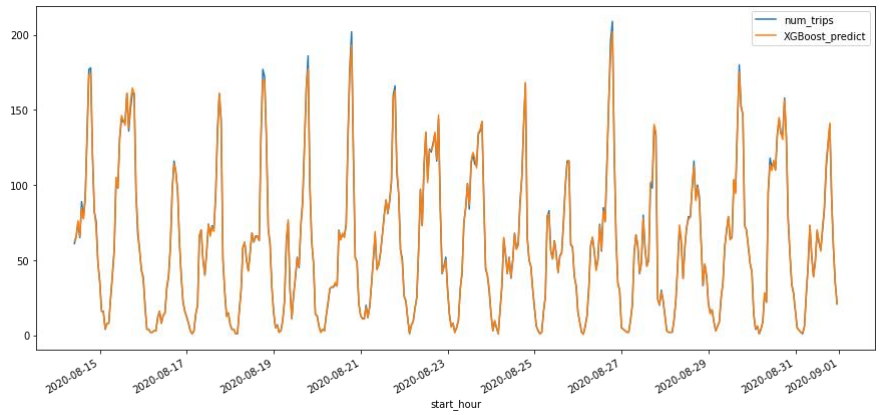

Figure 7 XGBoost prediction result

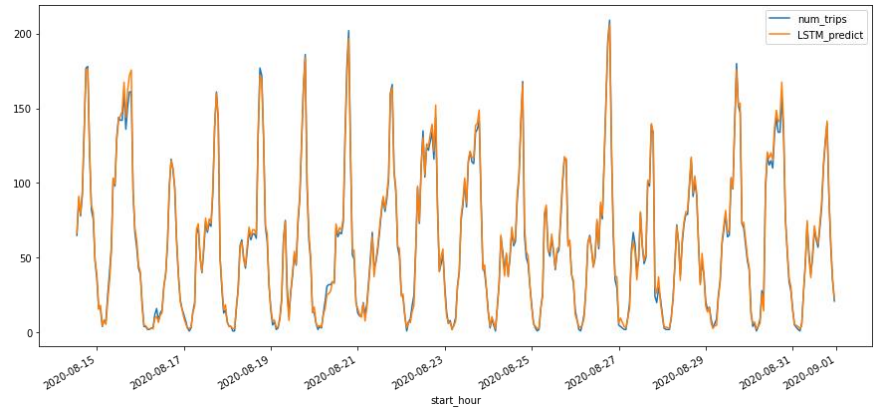

Figure 8 LSTM prediction result

The prediction results of XGBoost and LSTM are shown in figure 7 and 8. The orange line is the predicted hourly number of bike trips, and the blue line is the true number of bike trips. the predicted results of XGBoost and LSTM models are basically consistent with the trend of the actual value, but there is a small gap between the peak values. By comparison, LSTM has a larger gap in the peak value of sequence prediction, indicating that XGBoost has a higher prediction accuracy, which can be explained specifically through evaluation indicators.

6.3 Evaluation and Comparison

ARIMA and SARIMA models are used to predict daily bike demand, while XGBoost and LSTM models are used to predict hourly bike demand for the test data set, corresponding to the number of Citi bicycle trips between August 15, 2020 and August 31, 2020, as shown in Table 10. ARIMA and SARIMA are the control group. Because the hourly fluctuation of time series is difficult to conduct modeling, so the daily data is chosen for prediction.

For RMSE and MAE, the smaller the value, the smaller the prediction error, the more accurate the result, and the better the model. For R2, the larger the value is, the higher the fitting degree between the predicted value and the real value is, indicating a good prediction performance of the model.

Per Table 10, RMSE of the first two evaluation indexes of XGBoost model and LSTM model using grid search are both small, which are 1.53 and 3.09 respectively. They both have a very large R2, which means the models fit quiet well. In addition, XGBoost's training time is significantly shorter than LSTM's, though both took hours for training.

For ARIMA and SARIMA models, since they predict daily trips as XGBoost and LSTM predict hourly trips, it will lead to high RMSE and MAE values of ARIMA and SARIMA models, thus requiring normalization operations. However, RMSE and MAE values of ARIMA and SARIMA are also significantly higher than those of XGBoost and LSTM models, indicating that the prediction errors of the two models are large. The prediction accuracy of SARIMA model is slightly higher than that of ARIMA, indicating that the Citi bike travel data contains a certain seasonal component. Meanwhile, R2 of ARIMA and SARIMA is very low, indicating that the models have bad effects and the independent variable and the dependent variable have very little relationship. It is probably because they have omitted some other variables that have an impact on the dependent variable, showing the limitation of univariate prediction.Moreover, differencing method can also delete some important information of the data. Generally, the R2 is not high in ARIMA/SARIMA model. When accurate predictions are required, models with low R2 values should not be used. In summary, the results show that XGBoost model has the best prediction ability and fitting effect for the current trend.

Table 10 Evaluation of Models

Model Name	Frequency	MAE /Normalized MAE	RMSE /Normalized RMSE	R-squared
ARIMA	daily	184.04863 /0.21627	267.81608 /0.26937	0
SARIMA	daily	196.14556 /0.20526	241.13299 /0.25906	0.17400
XGBoost	hourly	0.89705 /0.01065	1.53669 /0.01383	0.99895
LSTM	hourly	2.39271 /0.02841	3.08978 /0.02781	0.99578

7 Conclusion and future work

In this paper, ARIMA, SRIMA, XGBoost and LSTM models are respectively applied to time series prediction of Citi bike trips per hour and per day during COVID-19 outbreak. In terms of data construction, ARIMA and SARIMA model only use daily citi bike sharing data in New York City from March 1, 2020 to August 31, 2020. The LSTM and XGBoost models use hourly Citi Bike data, weather data from the National Oceanic and Atmospheric Administration

(NOAA), and pandemic control data for New York City between March 1, 2020, and August 31, 2020. In the evaluation model, RMSE, MAE and R2 are used to compare the performance of the three models. After calculating the importance of features, feature selection is carried out on the data input into LSTM and XGBoost models. Through comparison, it is found that XGBoost model is superior to other models in all three indicators and has the highest prediction accuracy.

In time series prediction, ARIMA and SARIMA are the control group. Because the daily data is chosen for prediction,which is much short and can reduce the prediction accuracy to some extent. In addition, the input characteristics of ARIMA and SARIMA models are univariate, and the prediction is only made by the feature of the data itself, so the prediction has great limitations. The accuracy of SARIMA model is slightly higher than that of ARIMA, indicating that there are certain seasonal characteristics in time series data.

To improve the prediction accuracy of the model, XGBoost and LSTM model make prediction in the unit of hours. Limited by data size and single time step prediction, LSTM model has the characteristics of longest training time, medium prediction accuracy and medium performance. To improve the model training speed, the input XGBoost and LSTM data are integrated to some extent, and the site ID, longitude and latitude are deleted, while only nine features including user information, weather information and epidemic information are retained. In the actual prediction, the number of Citi bike trips frequently shows peaks and troughs in the unit of hour. LSTM and XGboost models can accurately predict the trend of the number of trips, but there are certain errors in the prediction of the specific values of the peaks and troughs. The prediction error of XGBoost model based on grid search optimization parameters is minimum, improving the prediction performance to a certain extent. This shows that there is room for further improvement of the model, and the quality and selection of data need to be improved in combination with other methods in the later stage. Because the predicted target is the number of Citi bike trips in New York City as a whole, the geographical characteristics of individual stations and the epidemic characteristics of large regions are excluded from the importance of features, so spatial dependence between stations is not considered.

In future studies, these problems can be gradually solved to better improve the prediction accuracy of Citi bike demand under the epidemic situation, so that epidemic prevention and traffic scheduling can be better designed after each policy of reopening. In the case of COVID-19, data sources have been helpful in predicting bike travel. The number and type of model factors can be adjusted appropriately, and other data related to the epidemic can be explored to improve the prediction performance of the model. Because of the clustering nature of the epidemic, the location of the Citi Bike station itself also has a large impact on the

prediction. Compared with City level prediction, it may be more accurate to model the travel pattern of a single station, which can ignore the spatial dependence between multiple stations and focus more the dispersion of epidemic transmission, thus making a more accurate prediction and providing more valuable reference for policy makers.

REFERENCES

[1] Konstantinos Pelechrinis, "Bike Sharing and Car Trips in the City: The Case of Healthy Ride Pittsburgh",University of Pittsburgh,October 17, 2016

[2] RanAna; ReneeZahnow, "Weather and cycling in New York: The case of Citibike", Journal of Transport, Geography, Volume 77, May 2019, Pages 97-112

[3] Yiyuan Lei ,Kaan Ozbay, "A robust analysis of the impacts of the stay-at-home policy on taxi and Citi
Bike usage: A case study of Manhattan ",Transport Policy 110 (2021) 487–498

[4] Zhaonan Qu SU, "Demand and Trip Prediction in Bike Share Systems",December 16, 2017

[5] Po-Chuan Chen, He-Yen Hsieh,"Prediction of Station Level Demand in a Bike Sharing System Using Recurrent Neural Networks",IEEE Conference on Vehicular Technology (VTC)

[6] Yexin Li, Yu Zheng, "Citywide Bike Usage Prediction in a Bike-Sharing System;IEEE Transactions on Knowledge and Data Engineering

[7] Haojie Li ,Yingheng Zhang,"Impacts of COVID-19 on the usage of public bicycle share in London",Transportation Research Part A 150 (2021) 140–155

[8] Haoyun Wang, Robert B. Noland,"Bikeshare and subway ridership changes during the COVID-19 pandemic in New York City",Transport Policy, Volume 106, June 2021, Pages 262-270

[9] YanPanaRay, ChenZheng,"Predicting bike sharing demand using recurrent neural networks",Procedia Computer Science,Volume 147, 2019, Pages 562-566

[10] Sathishkumar, VEJangwooPark,"Using data mining techniques for bike sharing demand prediction in metropolitan city",Computer Communications, Volume 153, 1 March 2020, Pages 353-366

[11] Divya Singhvi,1 Somya Singhvi,"Predicting Bike Usage for New York City's Bike Sharing System",Computational Sustainability: Papers from the 2015 AAAI Workshop

[12] Citi Bike Data: https://www.citibikenyc.com/system-data

[13] NOAA data:https://data.noaa.gov/dataset/

[14] NYC health data:https://www1.nyc.gov/site/doh/data/data-home.page

[15] NYC health policy data: https://www.health.ny.gov/health_care/medicaid/program/medicaid_health_homes/policy/index.htm

[16] ORACLE,"Time Series Concepts (oracle.com) Oracle8i Time Series User's Guide",Part Number A67294-01

[17] Robert Nau,"The mathematical structure of ARIMA models", Duke University

[18] Rob J Hyndman , George Athanasopoulos,"Forecasting: Principles and Practice (2nd ed) ,Monash University, Australia

[19] Mingda Zhang,"Time Series: Autoregressive models,AR, MA, ARMA, ARIMA",University of Pittsburgh

[20] Aayush Bajaj,"ARIMA & SARIMA: Real-World Time Series Forecasting",May 6th, 2021

[21] Dr. Jiban Chandra Paul, Md. Shahidul Hoque,"Selection of Best ARIMA Model for Forecasting Average Daily Share Price Index of Pharmaceutical Companies in Bangladesh: A Case Study on Square Pharmaceutical Ltd.,University of Chittagong, Bangladesh

[22] Jamal Fattah,"Forecasting of demand using ARIMA model",October, 2018

[23] E. Stellwagen,Len Tashman," ARIMA: The Models of Box and Jenkins",January 2013

[24] Enes Zvornicanin,"Choosing the best q and p from ACF and PACF plots in ARMA-type modeling",Baeldung on Computer Science,September 2021

[25] Aliyu Sani Aliyu ,"Application of Sarima Models in Modelling and Forecasting Monthly Rainfall in Nigeria",Asian Journal of Probability and Statistics

[26] James Bergstra,"Random Search for Hyper-Parameter Optimization",Universite de Montr ´ eal

[27] TianqiVhen, "XGBoost: A Scalable Tree Boosting System ",University of

Washington

[28] Shubham Malik,"XGBoost: A Deep Dive into Boosting",February 2020

[29] XGBoost Parameters — xgboost 2.0.0-dev documentation

[30] Ole-Edvard Ørebæk, Marius Geitle,"Exploring the Hyperparameters of XGBoost Through 3D Visualizations",Ostfold University Collage

[31] Sepp Hochreiter,"Long Short-term Memory",Neural Computation 9(8):1735-80,December 1997

[32] Xiaodan Zhu,"Long Short-Term Memory Over Recursive Structures",University of Ottawa

[33] Cheng-Hsin Yen,"Parameter Optimization for CNN-LSTM by Using Uniform Experimental Design",IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)

[34] S. Raschka,"Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning", November 2018

Page Number