International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 03 | Mar 2023 www.irjet.net p-ISSN: 2395-0072

Rides Request Demand Forecast- OLA Bike

Abdul Khayyum Farooqui

Research Scholar, Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology, Hyderabad 501301, India

---

Abstract: ***- The market shares and significance of ride-hailing or Transportation Network Companies (TNCs) like Uber, Lyft, and Ola are growing in numerous transportation markets. Big data technology and algorithms should be utilized to analyze the huge amounts of information accessible to enhance service reliability, estimate the efficiency of these systems, and assist them in meeting the demands of passengers. In this study, a novel model is developed using data from Ola, the leading ride-hailing service in Bangalore, to estimate the gap between rider demands as well as driver supply in a particular time period and specified geographic location. The data set used in this study was a ride request dataset. This dataset would have the following attributes: ride booking time, pickup location, and drop point latitude-longitude. The number of data points related to ride requests are, the columns of the data are Id of the customer, timestamp booking, pickup latitude, pickup longitude, drop latitude and drop longitude. On the phone-based Ola application, a passenger "calls a ride," (makes a request) by inputting the point of origin and destination and tapping the Request Pickup button. A driver answers the request by taking the order. Even while the training set is little in comparison to the whole Ola ride-hailing industry, it is large enough for patterns to be found and generalized.***

Keywords**: -*****Big Data; Ride-Hailing, K-means Clustering; XGBoost; Root Mean Squared Error***

1.INTRODUCTION

The ride-hailing (Ola) service sector has been expanding for a few years, and it is anticipated to continue expanding in near future. Ola drivers must decide where to wait for passengers since they may arrive rapidly. Additionally, passengers like an immediate bike service whenever required. People who have issues with booking Ola bikes, which sometimes cannot be fulfilled or the wait time for the arrival of the trip is particularly lengthy owing to the lack of a nearby Ola bike. If you successfully reserve an Ola bike in one go, consider yourself fortunate[1].

Ola is acquiring a greater market share and significance in a variety of transportation markets. Big data technologies and algorithms should be employed to handle the enormous amounts of information that are available to enhance service efficiency[2]. This will allow for more accurate estimates of efficiency as well as assistance in meeting the needs of riders[3]. This work develops a model to forecast supply and demand mismatches using information from the leading ride-hailing company in Bangalore. The percentage of Indians who travel by taxi, bus, or rail is among the highest in the world and few of the Indians 1.4 million residents own automobiles [4]. The leading ride-hailing business in Bangalore, Ola, handles more than 1 lakh rides daily and gathers more than 5GB of data.

It has become important for Ola (and other e-haling) company to forecast the demand for their Ola bikes so that they may better understand that demand and maximize the efficiency of their fleet management.

A novel model based on users' ride request dataset is proposed to address these problems; it would include characteristics such as ride booking time, pickup place, and drop point latitude-longitude. This model would predict demand for a certain period in various city areas, assisting the business in maximizing the density of Ola bikes to meet consumer demand.

The rest of this paper is organized as follows. Section 2 discusses about the related work. In Section 3 the Background is presented. Methodology of the study is presented in Section 4. Section 5 shows the obtained results and discussion. The paper is concluded in the Section 6.

2.RELATED WORK

[5] Had examined the problem of attempting to forecast the supply-demand gap in ride-sourcing services over the near term. In contrast to the previous studies, which divided a city area into a number of square lattices, this study divided the city area into a number of regular hexagon lattices. This difference in approach was motivated by the fact that hexagonal segmentation has an unambiguous neighborhood definition, a smaller edge-to-area ratio, and isotropy. The study proposed three hexagon-based convolutional neural networks (H-CNN), the input and output of which are multiple local hexagon maps, in order to capture the spatiotemporal properties in a hexagonal fashion. A hexagon-based ensemble technique is developed to enhance prediction performance. The H-CNN models are determined to greatly beat the benchmark algorithms in terms of

accuracy and robustness after being validated with a 3-week real world ride-sourcing dataset in Guangzhou, China.

[6] had developed a model for ride hailing demand forecasting that was based on deep learning in an effort to reach high levels of accuracy when dealing with challenges of a similar kind. This also addressed a constraint that is present in previous models for predicting ride hailing demand, which is that the region is organized into a rectangle grid, and all travel demand projections are performed within rectangular cells, rather than inside city neighborhood zones. The suggested model estimates demand for travel between city neighborhood zones. The proposed model outperforms the CNN and LSTM models up to 18.41% in RMSE and 22.65% in R^2 values, according to trials using a real-world rental car dataset in New York City.

[7] had provided several machine learning algorithms in order to characterize and forecast the demand for on-demand ride-hailing services in the near future. The spatio-temporal estimate of demand, which is a function of variable effects relating to traffic, price, and environmental factors, was also proposed. In terms of the methods, a single decision tree, bootstrap-aggregated (bagged) decision trees, random forest, boosted decision trees, and an artificial neural network for regression have all been adapted and systematically compared using a number of different statistics, such as R-square, Root Mean Square Error (RMSE), and slope. With an aggregated-time interval of ten minutes, 199,584 time-slots that describe the spatio-temporal ride-hailing demand have been extracted from the data. On the basis of two independent samples from this dataset, all techniques are trained and validated. The findings showed that boosted decision trees, artificial neural networks, random forests, bagged decision trees, and single decision trees all provide the greatest prediction accuracy while minimizing the danger of over-fitting.

3.BACKGROUND

3.1 Mini-batch k-means clustering

When clustering on enormous datasets, the Mini-batch K-means clustering technique is an alternative to the K-means algorithm. Because it does not cycle over the complete dataset, it sometimes outperforms the traditional K-means method when dealing with large datasets. It generates random batches of data to be kept in memory, and then gathers a random batch of data during each iteration to update the clusters. The Mini-batch K-means algorithm's primary benefit is that it reduces the computing cost (time) of finding a cluster. Although the K-means method is also used, but when working on a huge dataset, the mini-batch approach is utilized.

3.2 Multi-step time series forecasting

Predicting a succession of values in a time series is known as multistep-ahead prediction. A common strategy, known as multi-stage prediction, involves applying a predictive model step-by-step and using the anticipated value of the current time step to calculate its value in the next time step. Multi-step forecasting is useful where the field of application requires long-term duration forecasting[8].

Predicting the subsequent H values $\left[yN_{obs}+1,\ldots,yN_{obs}+H\right]$ of a historical time series $\left[y_1,\ldots,yN_{obs}\right]$ made up of N_{obs} observations is known as a multi-step ahead (also known as long-term) time series forecasting problem. where H > 1 denotes the forecasting horizon.

There are five different types of computation methods for forecasting multiple steps in ahead. In this study, recursive approach is taken into consideration.

3.2.1 Recursive strategy

The Recursive (also known as Iterated or Multi-Stage) technique is the most traditional and logical method of forecasting. In this method, a single model f is trained to carry out a one-step forecast[9].

$$y_{t+1}=f\left(y_t,\ldots,y_{t-d+1}\right)+w \qquad t\in\{d,\ldots,N-1\} \tag{1}$$

When predicting H steps in advance, start by applying the model to the first step. Use the value you just predicted as one of the input factors for predicting the next action after that (using the same one-step ahead model). Continue in this way until the whole horizon has been forecasted.

Let $\hat{f}$ represent the trained one-step-ahead model. The forecasts are then given by

$$\hat{y}_{N+h}=\begin{cases}\hat{f}\left(y_N,\ldots,y_{N-d+1}\right) & if\ h=1\\ \hat{f}\left(\hat{y}_{N+h-1},\ldots,\hat{y}_{N+1},y_N,\ldots,y_{N-d+h}\right) & if\ h\in\{2,\ldots,d\}\\ \hat{f}\left(\hat{y}_{N+h-1},\ldots,\hat{y}_{N+h-d}\right) & if\ h\in\{d+1,\ldots,H\}\end{cases} \tag{2}$$

The recursive technique may perform poorly in multi-step forward forecasting jobs, depending on the amount of noise in the time series and the forecasting horizon. In fact, this is particularly true if the embedding dimension d is greater than the forecasting horizon h, since at that point all the inputs are predicted values rather than real observations (Equation 2). The Recursive method is sensitive to the accumulation of mistakes with the forecasting horizon,

which is the cause of the probable inaccuracy. As these projections are used to inform later forecasts, any errors existing in intermediate forecasts will be propagated forward. Despite these drawbacks, the Recursive approach has been successfully used to predict a variety of real-world time series utilizing various machine learning models, such as nearest-neighbors and recurrent neural networks.

4.METHODOLOGY

Ride hailing companies (such as Ola) are losing money and market share to their competitors, due to their failure to satisfy the trip demands of many consumers. To solve this issue, a novel model is presented out to forecast the demand for rides in a particular area and during a certain time period.

Dataset: The data set used in this study was a ride request dataset. This dataset would have the following attributes: ride booking time, pickup location, and drop point latitude-longitude. The number of data points related to ride requests are, the columns of the data are Id of the customer, timestamp booking, pickup latitude, pickup longitude, drop latitude and drop longitude.

Every user has a unique customer id, the booking timestamp is the date and time of the ride booking (IST time), the pickup latitude is the ride request pickup latitude, the pickup longitude is the ride request pickup longitude, the drop latitude is the ride request drop latitude, and the drop longitude is the ride request drop longitude.

Data Preparation: In order to create a prediction model for the demand for rides in a certain area at a given time, the data must first be preprocessed to determine the actual estimated demand by consumers. I eliminated requests for rides that were very likely to be problematic in order to evaluate the genuine demand.

- If multiple bookings are made from the same latitude and longitude within 'h' hours of the most recent booking, only count the first ride request made by the user.
- Consider just one ride request from a user if there are more than one within 'm' minutes of the most recent booking time (latitude and longitude may or may not be the same)
- If the geodesic distance between the pickup and drop-off points is less than 50 meters, the transportation request should be regarded as fraudulent.
- Consider the following ride requests as a system error: ["6.2325274", "35.6745457", "68.1113787", "97.395561"] All requests where the pick-up or drop-off location is beyond the bounds of Bangalore.
- We don't want to provide intercity rides or lengthy bike excursions; thus, we remove such services if the geodesic distance between the pick-up and drop-off points is more than 500 km.

Clustering Regions (pickup latitude and longitude) with Mini-Batch K-means

Due to the fact that geographical data cannot be used for demand forecasting activities, geospatial engineering was necessary. Given 4 million data points, using standard K-means for clustering would take hours of computing time. Consequently, via a technique known as "Mini Batch K-Means Clustering," we have subdivided the whole of Bangalore into 50 distinct zones.

Mini-Batch-K-Means is a variation of the K-Means method that still aims to optimize the same objective function while using mini-batches to speed up processing. In each training cycle, mini-batches, which are subsets of the input data, are randomly picked. By using these mini-batches, the amount of computation needed to get a local solution is dramatically reduced. Mini-batch k-means, in contrast to other methods that speed up k-means convergence, yields results that are often only marginally inferior than those of the standard approach.

Data Features

The resultant features are customer Id (unique Id given to each user), booking timestamp (booking timestamp of ride IST) and pick up cluster Id (this is computed by clustering over pickup latitude and longitude).

To determine the number of demand/ride requests from a region: In order to collect the number of customer ids that booked trips from those areas during that timeframe, we split time into 30min intervals, creating a total of 24Hours * 2 = 48 (30min intervals).

Multi-Step Time Series Forecasting

The challenge of anticipating a succession of values in a time series is known as multistep-ahead prediction. Applying a predictive model step-by-step and using the anticipated value of the current time step to calculate its value in the next time step is a common strategy known as multi-stage prediction.

Test the signal's self-similarity across a range of delay durations by performing an auto-correlation test. Knowing whether to use latency features is helpful.

On experimenting and modeling it was observed: For forecasting best set of features were:

- pickup_cluster: region_id computed by clustering of pickup lat-long.
- mins: Booking time broken into 30mins intervals
- hour: the hour of booking the ride
- month: booking month
- dayoftheweek: day of the week when the ride was booked; weekends demand is less than weekdays
- lag_feature of 7 steps: previous demands as input (7 is used by analyzing autocorrelation and partial autocorrelation plots).
- rolling mean of last 7 steps: sliding window mean of last 7 ride demands

Train test split:

The train test split method is used to split the data into train and test sets. First, we need to divide the data into features (X) and labels (y). The data-frame gets divided into X_train, X_test, y_train and y_test. X_train and y_train sets are used for training and fitting the model. The X_test and y_test sets are used for testing the model if it's predicting the right outputs/labels. we can explicitly test the size of the train and test sets.

Train set: The training dataset is a set of data that was utilized to fit the model. The dataset on which the model is trained. This data is seen and learned by the model.

Test set: The test dataset is a subset of the training dataset that is utilized to give an accurate evaluation of a final model fit.

Validation set: A validation dataset is a sample of data from your model's training set that is used to estimate model performance while tuning the model's hyper-parameters.

Extreme gradient boosting (XGBoost)

Extreme gradient boosting, also known as XGBoost, is an algorithm that is built on top of a gradient boosting tree and has the potential to significantly contribute to the process of gradient improvement. When it comes to solving issues involving regression and classification, XGBoost, which is based on the theory of classification and regression trees, is a highly effective solution. In addition, XGBoost also stand for a soft computing library that integrates GBDT techniques together with the recently developed algorithm. After optimization, the objective function of XGBoost is composed of two distinct elements, which reflect the deviation of the model and the regular term to avoid over-fitting. These parts work together to achieve the goal of maximizing the predictive power of the algorithm. $D=\{(x_i,y_i)\}$ is a notation for a data set that has n samples and m features, and in which the predictive variable is an additive model that is composed of k basic models. The data set has n samples and m features. The outcomes of the sample prediction are as detailed in the following:

$$\hat{yy}_i=\sum_{k=1}^{K} f_k(xx_i),\ f_k\in\varphi \tag{3}$$

$$\varphi=\{f(xx)=w_s(xx)\}(s:R^m\to T, w_s\in R^T) \tag{4}$$

Where $\hat{yy}_i$ stands for the prediction label, xx_i for one of the samples, $f_k(xx_i)$ for the projected score, and φ for the set of the regression tree, which is a tree structure with the parameters s, $f(xx)$, and w representing the weight of leaves and the number of leaves, respectively.

In XGBoost, the objective function takes into account both the conventional loss function and the complexity of the model. It is possible to utilize it to assess the algorithm's operating efficiency and effectiveness. The first term in Formula (5) reflects the conventional loss function, while the second term in that formula indicates the complexity of the model.

$$obj=\sum_{i=1}^{m} l\left(yy_i,\hat{yy}_i^{(t-1)}+f_i(xx_i)\right)+\Omega(f_{kk}) \tag{5}$$

$$\Omega(f_{kk})=\gamma T+1/2\lambda w^2 \tag{6}$$

Here i is the number of samples in the dataset and m denotes the total amount of data fed into the k^{th} tree in both of these calculations. Additionally, the complexity of the tree is adjusted using γ and λ. The final learning weight may be smoothed using regularization to prevent over-fitting.

Model evaluation:

RMSE:

The Root Mean Squared Error (RMSE) is one of the two main performance indicators for a regression model. It measures the average difference between values predicted by a model and the actual values. It provides an estimation of how well the model is able to predict the target value (accuracy).

$$RMSE=\sqrt{\frac{1}{w}\sum_{i=1}^{N} w_i u_i^2} \tag{7}$$

where:

SSEw = Weighted Sum of Squares

W = Total weight of the population

N = Number of observations

wi = Weight of the i-th observation

ui = Error associated with the i-th observation

5.RESULTS AND DISCUSSION

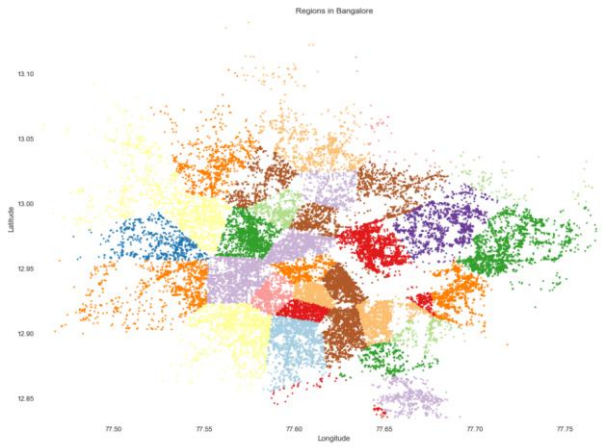

Fig-1: Clustering with mini batch k-means

The above figure shows the post clustering region division for Bangalore, the mini batch k-means clustering algorithm is used to divide the required cluster heads. The clustering of latitude and longitude is completed and the clusters divided into 50 pickup clusters.

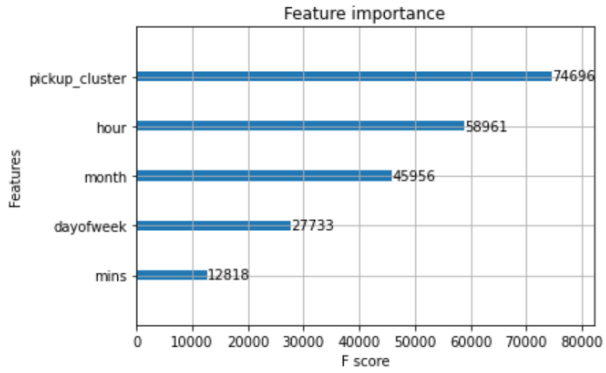

Fig-2: feature importance

The above figure shows the importance of the features, the pickup cluster feature score is 74696, hour score is 58961, month score is 45956, day of week score is 27733 and minutes score is 12818.

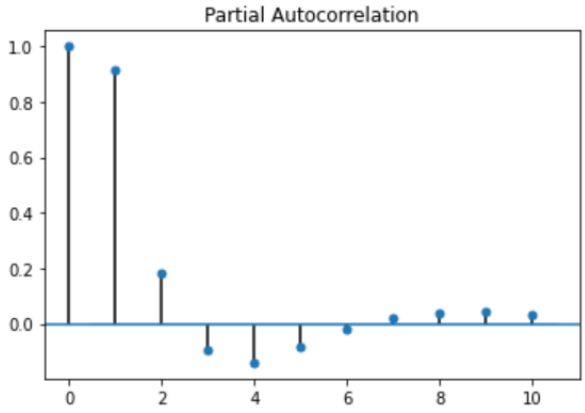

Fig-3: Partial Autocorrelation

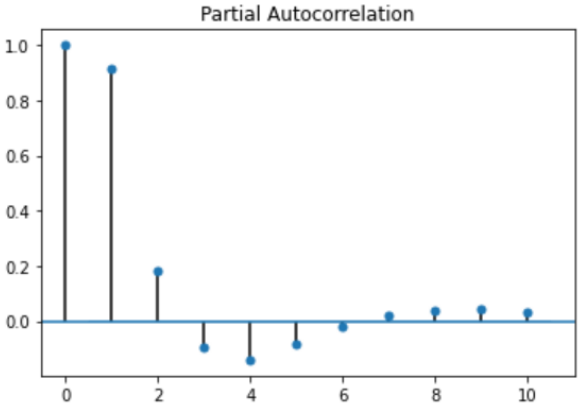

Fig-4: Partial Autocorrelation

The above figures 3 and 4 shows the partial autocorrelation, it is a measure of the correlation between time series with a lagged version of itself after eliminating the variations already explained by the intervening comparisons, from the figures it is observable that the partial autocorrelation function shows a high correlation with the first two lag and lesser correlation with 3rd and 4th lag.

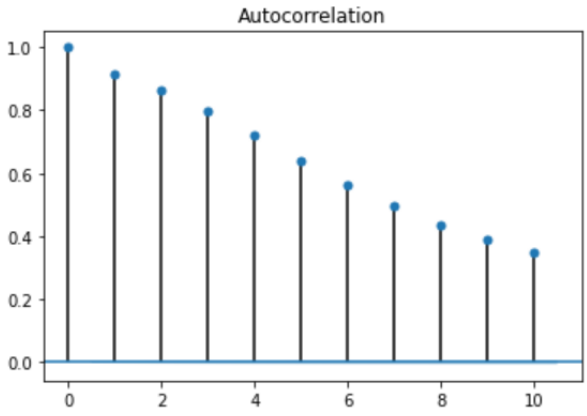

Fig-5: Autocorrelation

The above figure 5 is the Autocorrelation function plot, it is a measure of the correlation between the time series and the lagged version of itself. From the figure it is observable that the autocorrelation function shows a slow decay, which means that the future values have a very high correlation with its past values.

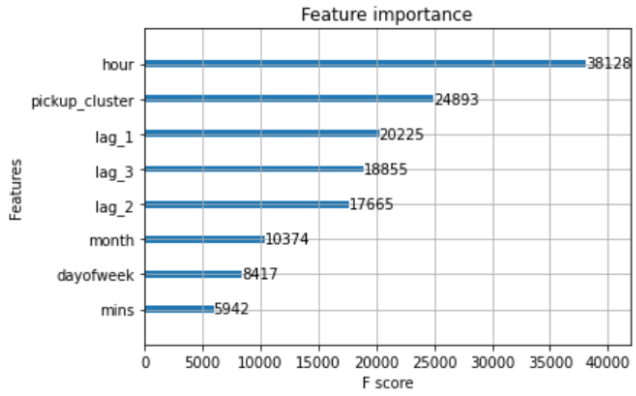

Fig-6: Feature importance

The above figure shows the importance of the feature with Random-Forest, from the figure it is observable that the hour feature score is 38128, pickup cluster feature score is 24893, lag1 score is 20225, lag 2 score is 17665, lag 3 score is 18855, month score is 10374, day of week score is 8417, and minutes score is 5942.

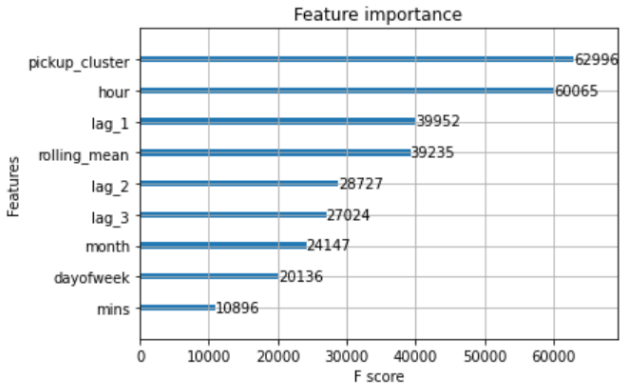

Fig-7: Feature Importance Graph

The above figure shows the importance of the feature in XGBoost, from the figure it is observable that the pickup cluster feature score is 62996, hour feature score is 60065, lag1 score is 39952, rolling mean score is 39235, lag 2 score is 28727, lag 3 score is 27024, month score is 24147, day of week score is 20136, and minutes score is 10896.

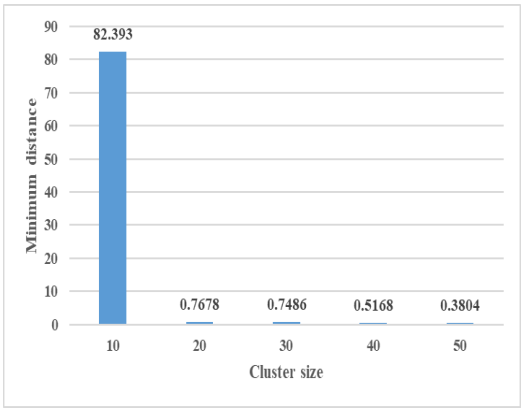

Fig-8: Inter Cluster Distance

The above figure 6 shows the inter cluster distance between two clusters, the inter cluster distance required is less than 0.5miles. From the above figure it is observable that the inter cluster distance for 50 clusters is 0.3804, so the number of clusters considered in this work are 50.

6.CONCLUSIONS

In order to handle the issue of ride demand forecasting, a novel XGBoost regressor model is proposed in this work. The data preprocessing, geospatial engineering methods are utilized to convert latitude and longitude, to cluster Id using Mini-Batch Kmeans algorithm, and then multi-step forecasting is used to forecast the demand for ride requests coming from an area at a certain time. The proposed XGBoost Regressor model score is 0.916, and the RMSE values for train and test are 2.287 and 2.456.

REFERENCES

[1] J. Ke, H. Zheng, H. Yang, and X. (Michael) Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transp. Res. Part C Emerg. Technol.*, vol. 85, pp. 591–608, 2017, doi: 10.1016/j.trc.2017.10.016.

[2] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," *WSDM 2018 - Proc. 11th ACM Int. Conf. Web Search Data Min.*, vol. 2018-Febuary, no. February, pp. 736–744, 2018, doi: 10.1145/3159652.3159682.

[3] G. Cantelmo, R. Kucharski, and C. Antoniou, "Low-Dimensional Model for Bike-Sharing Demand Forecasting that Explicitly Accounts for Weather Data," *Transp. Res. Rec.*, vol. 2674, no. 8, pp. 132–144, 2020, doi: 10.1177/0361198120932160.

[4] C. Guido, K. Rafal, and A. Constantinos, "A low dimensional model for bike sharing demand forecasting," *MT-ITS 2019 - 6th Int. Conf. Model. Technol. Intell. Transp. Syst.*, 2019, doi: 10.1109/MTITS.2019.8883283.

[5] J. Ke *et al.*, "Hexagon-Based Convolutional Neural Network for Supply-Demand Forecasting of Ride-Sourcing Services," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4160–4173, 2019, doi: 10.1109/TITS.2018.2882861.

[6] Z. Ara and M. Hashemi, "Ride hailing service demand forecast by integrating convolutional and recurrent neural networks," *Proc. Int. Conf. Softw. Eng. Knowl. Eng. SEKE*, vol. 2021-July, no. Ml, pp. 441–446, 2021, doi: 10.18293/SEKE2021-009.

[7] I. Saadi, M. Wong, B. Farooq, J. Teller, and M. Cools, "An investigation into machine learning approaches for forecasting spatio-temporal demand in ride-hailing service," 2017, [Online]. Available: http://arxiv.org/abs/1703.02433

[8] C. Wang, Y. Hou, and M. Barth, "Data-Driven Multi-step Demand Prediction for Ride-Hailing Services Using Convolutional Neural Network," *Adv. Intell. Syst. Comput.*, vol. 944, pp. 11–22, 2020, doi: 10.1007/978-3-030-17798-0_2.

[9] S. Ben Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7067–7083, 2012, doi: 10.1016/j.eswa.2012.01.039.