Heliyon 9 (2023) e20129

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Scalability evaluation of forecasting methods applied to bicycle sharing systems

Alexandra Cortez-Ordoñez [a], Pere-Pau Vázquez [b], José Antonio Sanchez-Espigares [c,*]

[a] Department of Statistics and Operations Research, UPC-BarcelonaTECH, Avda. Diagonal, 647, Planta 6, 08034 - Barcelona, Spain
[b] ViRVIG Group Department of Computer Science, UPC-BarcelonaTECH, C/ Jordi Girona 1-3, Ed Omega 137, 08034 - Barcelona, Spain
[c] Department of Statistics and Operations Research, UPC-BarcelonaTECH, Avda. Diagonal, 647, Planta 6, 6-67, 08028 - Barcelona, Spain

ARTICLE INFO

Keywords:
Forecasting methods
Bike demand forecasting
Bike sharing systems

ABSTRACT

Public Bicycle Sharing Systems (BSS) have spread in many cities for the last decade. The need of analysis tools to predict the behavior or estimate balancing needs has fostered a wide set of approaches that consider many variables. Often, these approaches use a single scenario to evaluate their algorithms, and little is known about the applicability of such algorithms in BSS of different sizes. In this paper, we evaluate the performance of widely known prediction algorithms for three sized scenarios: a small system, with around 20 docking stations, a medium-sized one, with 400+ docking stations, and a large one, with more than 1500 stations. The results show that Prophet and Random Forest are the prediction algorithms with more consistent results, and that small systems often have not enough data for the algorithms to perform a solid work.

1. Introduction

The exponential growth in both the popularity and number of public bicycle sharing systems (BSS) has transformed the urban mobility patterns. Currently, there are more than 1700 BSS [1] of different types (public, private, dockless …) around the world. Major or large cities such as New York, Paris, London, or Barcelona have deployed BSSs to complement public transportation means. Besides, the recognition of its multiple health benefits and its successful support for transport connection has led to many small and medium cities around the world, to also adopt BSSs.

These systems provide citizens facilities to move around the city. In most of BSS, citizens can rent bikes from the nearest docking station and return them to other station. Since public BSSs are usually composed by a fixed number of docking stations and bikes, its continuous usage can create unbalanced bikes distribution. This creates two types of problems: the lack of bikes at peak hours, and the inability to drop off bicycles in certain docking stations because no free slots are available. Both problems may lead to customer loss. Therefore, a relevant challenge for system operators and managers consists in adequately perform balancing operations by repositioning bikes among stations.

On the other hand, small cities are typically less capable of carrying out such operations, because of their limited resources. Besides, system operator and managers of BSS in these cities are strongly interested into getting detailed insights on how the system behaves as a whole. Therefore, a short-term prediction of the number of bicycles that will arrive and depart from each docking

* Corresponding author.
E-mail addresses: alexandra.cortez@upd.edu (A. Cortez-Ordoñez), pere.pau.vazquez@upc.edu (P.-P. Vázquez), josep.a.sanchez@upc.edu (J.A. Sanchez-Espigares).

https://doi.org/10.1016/j.heliyon.2023.e20129
Received 20 March 2023; Received in revised form 11 September 2023; Accepted 12 September 2023

Available online 19 September 2023

station may be highly useful to assist system monitors to optimize balancing operations. Opening this information to BSS customers may also help citizens to plan their bike trips in advance.

The optimization of rebalancing operations and methods to improve the accuracy of demand prediction models have been addressed by different authors and several approaches has been proposed. These studies mainly focus on the optimization of larger BSSs, usually using one city to test the performance of one or more machine learning algorithms. Unfortunately, to the authors' knowledge, no studies have been developed to assess the main differences between BSSs with different size, characteristics, or usage patterns, and how forecasting algorithms will perform in each case.

In this study, we aim to fill this gap in the literature. By using data from three cities: Logroño (Spain), Barcelona (Spain), and New York (US) which have BSSs of significant different sizes, we compare the performance of different prediction models. We have tested five, widely used machine learning algorithms to predict short-term bikes arrivals and departures at station level: ARIMA, Linear Regression, Random Forest, Prophet, and XGBoost. Then, we evaluate model performance and results for each algorithm and BSS. The contributions of our paper are twofold:

- An analysis of five machine learning algorithms to predict short-time station-level arrivals and departures in BSS.
- A performance comparison of these prediction algorithms for BSSs of different characteristics and sizes (small, medium, and large).

The results show that BSS usage behavior is closely related with the system size, and how this usage dynamic impacts prediction models' scalability. In the case of the selected medium size system (Barcelona), Random Forest outperformed other models when predicting arrivals and departures. For the small and large system, Prophet had a better performance. Additionally, an interesting relationship was found between the usage ratio and the error metric. In the small size BSS this relationship is linear and positive, for the medium BSS it is linear and negative, and the large BSS has a non-linear and negative relationship. This implies that the higher the usage ratio, the lower is the error in the medium and large BSS. For the three cities considered, the docking stations located in the city center have also the higher usage ratio, and the error depends on the relationship previously mentioned. Moreover, in New York and Barcelona the error levels increase during the first interval of each day, when the usage ratio is lower, while in Logroño there is no clear pattern. Finally, we showed that the error levels, especially in Barcelona, are similar to the ones found by other authors and in Kaggle competitions.

The rest of the paper is organized as follows: Section 2 describes related work. Details about the selected bike sharing systems are given in Section 3. Section 4 contains data compilation and processing, as well as a description of the algorithms and error metrics that have been used. Model results are presented in Section 5. A discussion of the main results can be found in Section 6. Finally, Section 7 contains the conclusions and recommendations for future research.

2. Previous work

Bike Sharing Systems have received increasing attention in the latest years thanks to their associated benefits, such as pollution reduction, improvement of citizens' health, new means of transportation, and is also considered a mean to get more sustainable cities [2–4]. Many cities across the world are adopting public and private BSSs to provide their citizens a convenient, low-cost and environmental-friendly mean of transportation. Citizens have welcomed these systems since they also offer them multiple benefits such as avoidance of traffic congestion, lower costs, or fewer responsibilities associated with owning a bicycle (maintenance, theft, or storage). However, together with their popularity, challenges have also increased, especially those related to civic behavior (e.g., vandalism), the special characteristics of a city (e.g., elevation) [5], or how to achieve access equity [4].

Since the usage of these systems tends to be highly irregular, particularly at peak hours, system managers and operators have the challenge to optimize balancing operations between stations by predicting the rental demand of bikes. Accurate forecast models may help them to generate efficient vehicle routes [6,7] or different possibilities of smart traffic control [8]. It is also important for citizens to plan their trips in advance and select a station to pick-up or drop-off a bike [9].

During the last years, notable efforts have been put on the analysis of how the BSSs are being used. Traffic patterns have been studied by many authors [10–12] as well as the impact of destination preferences in the usage flows [13,14]. Urban configuration and its influence in bike's demand has also been an important area of study. For instance, Kim et al. [15] analyze how the elevation of stations may affect the trips. Frade and Ribeiro [16] also consider the elevation of the neighborhoods in Coimbra to propose a demand model. The characteristics of trips [17–19] and the effect of important events [11], weather [20,21] and calendar events [21–23] have become another important line of investigation. These studies have mainly focused on identifying the factors that influence bike sharing flows (demographic, meteorological, population, etc.) for one city. But, we are also interested in how these factors contribute to improve the prediction of short-term arrivals and departures at station-level in different BSSs.

The development of forecasting models for BSSs has been possible thanks to the massive data generated, especially by large BSSs. Some authors have focused on the prediction of trip destination and trip duration [24]. Additionally, different machine learning models have been tested to predict bike demand, such as spatial generalized ordered response [10], linear regression [25,26], Multiple Additive Regression Trees [24], among others. Novel approaches such as the Quantum Bayesian Networks (QBN) have also been tested [27]. These investigations mainly focus on the accuracy of one model for one BSS, but the comparison of the performance of different models with different BSSs has not been studied.

For larger systems, it makes sense to create clusters and analyze them collectively to get insights. Many studies have analyzed different forms to group stations based on metrics such as the *activity score*, as was performed by Froehlich et al. [28], similar behavior

of trips [29], availability levels [9], grouping by day of the week or month [30], among others. Cluster-level demand forecast has also been investigated by different authors. Li et al. [31] clustered docking stations and applied Gradient Boosting Regression Tree to predict bike demand. Opposite to these studies, we aim to predict station-level arrivals and departures and compare the efficiency of various models with data from BSSs with different characteristics.

Several algorithms for station-level demand forecast have been investigated in the latest years. Lin et al. [32] used data from New York and a Graph Convolutional Neural Network to predict hourly demand in each station. Chen et al. [33] also used New York BSS data and proposed a Recurrent Neural Network (RNN) to predict station-level demand. We are also interested in the analysis of station-level demand. However, we propose a comparison of different algorithms to select the most accurate. Data from Montreal was used to test different features and provide station-level arrivals and departures demand [34]. Similarly, Hu et al. [35] have improved the accuracy of station-level demand prediction using feature engineering with data from New York BSS. We will focus on the performance comparison of different prediction algorithms in systems with different sizes, rather than the features that influence the bike sharing demand.

Some authors have studied different prediction algorithms and compare their performance to select the more efficient. Cortez and Vázquez [36] predicted bike demand in Barcelona (Spain) using four machine learning models. Nevertheless, their focus was to provide a visual tool for both system operators and users, rather than test the accuracy of the models. Tomaras et al. [37] have proposed a combination of a Gradient Boosted Regression Tree, and a Holtz's model to create a tool called SmartBIKER that assist system operators with rebalancing. The authors tested their proposal using New York data. However, their focus is on the improvement of the tool rather than the performance of the prediction methods. Yin et al. [38] have analyzed four algorithms with Washington, D.C. data. Feng et al. [39] have also used Washington, D.C. data to demonstrate the better performance of Random Forest model over conventional multiple Linear regression. Wang and Kim [40] also tested three models, Random Forest, Long Short-Time Memory (LSTM) and Gated Recurrent Units (GRU) with data from Suzhou (China). Their results showed no significant difference between the accuracy achieved by the models tested. Similarly, Choi et al. [41] used data from Seoul, Korea and found small differences in the performance of Decision Tree (DT), Random Forest (RF), and Extra Tree (ET) models. Xu et al. [42] proved that LSTM algorithm has provided better results for Nanjing (China). Hulot et al. [43] used data from Bixi, Montreal to test several models and concluded that Gradient Boosted Tree achieves better scores than Random Forest and MultiLayer Perceptron (MLP). The performance of predictive models in a small BSSs has been analyzed by Lozano et al. [44]. These authors found that Random Forest outperform other models in Salamanca (Spain) BSS. We share with these studies the goal of comparing different algorithms to identify the one which provides a more accurate prediction. However, we also focus on the performance of these algorithms in systems with different characteristics and sizes.

Demand prediction and comparison in different systems is a work in progress. Rixey [45] performed a regression analysis to identify the more significant variables to predict station-level demand in three cities of US. Li et al. [31] have also tested one prediction model in New York and Washington, D.C. systems. However, these BSSs are of similar characteristics and no results' comparison was possible since authors only used one model to predict bike demand.

For more details, the above-mentioned studies are summarized in Table 1.

3. BSS characteristics

The increasing availability of bike-sharing data has created many opportunities to analyze information about different systems and in multiple domains. In this study, we are interested in a multi-city comparison of BSSs with different sizes and characteristics, focusing on the behavior and consistency of forecasting models. For this purpose, we have selected three cities with different BSS sizes: Logroño in Spain (small), Barcelona in Spain (medium), and New York City in US (large). The characteristics of these systems are described as follows:

3.1. Logroño - BiciLog

Logroño is a small city in the north of Spain with a population of more than 150 000 inhabitants. Its BSS, called BiciLog, was inaugurated in 2018. Today, the system has 23 docking stations, about 300 mechanical bicycles and almost 3 000 subscribers. Tourist and permanent residents can make use of the system as they have options for daily, weekly, monthly and yearly subscriptions. The price varies between 5 euros per day and 36 euros per year.

3.2. Barcelona - Bicing

Barcelona is known as one of the most touristic cities in Spain, with a population of more than 1.6 million of inhabitants. Its BSS named Bicing was opened in 2007. Nowadays, Bicing has 519 docking stations, 5 000 mechanical bikes, 2 000 electrical, and more than 130 000 subscribers. Users can rent a bike and drop-off the bike in any station, as all can handle electrical or mechanical bikes. Bicing is a system designed only for permanent residents, with two service options, where prices vary between 35 and 50 euros per year.

3.3. New York - Citi Bike

Located in US, New York has a population of more than 8.8 million of inhabitants. Its BSS, called Citi Bike, began operations in 2013. Currently, it has more than 1 500 docking stations, about 24 500 bikes and more than 12 million of subscribers. The plans

Table 1
Literature review summary.

Study	BSS	Features used	Model	Results (best model)
[10]	Washington, U.S.	customers characteristics (age, work status, gender, number of automobiles, number of children, and income)	Spatial generalized ordered-response model	MAPE within 10%
[24]	Chicago, U.S.	user and stations' characteristics, and time based features	Multiple Additive Regression Trees (MART) model	accuracy around 87%
[25]	Malmö, Sweden	the absolute number of bicycles, and the deviation from a long-term trend estimation of the expected number of bicycles, day of week, national holidays, school breaks, and bridge days	Linear regression model, Rep tree, Multi-layer Perceptron, and Support Vector Regression (SVM) with 2nd and 3rd degree polynomial kernels	relative error to around 30% for the SVM model
[26]	Malmö, Sweden	the absolute number of bicycles, and the deviation from a long-term trend estimation of the expected number of bicycles, day of week, national holidays, school breaks, and bridge days	Support vector machine, Linear Regression, Ridge Regression, Lasso and Bayesian Ridge Regression	R2 coefficient of 0.677 for Bayesian Ridge Regression
[27]	New York, U.S	mobility data	Quantum Bayesian Networks (QBN)	RMSPE within 2%
[28]	Barcelona, Spain	prediction window, time of the day, amount of historical data and stations clusters	Last value, Historic mean, Historic Trend and Bayesian Network	Similar behavior of methods with an error of around 13%
[31]	New York and Washington, U.S.	bike usage, time and weather factors	Gradient Boosting Regression Tree (GBRT)	RMSLE around 0.37
[32]	New York, U.S	stations' availability	Graph Convolutional Neural Network (GBRT)	RMSE around 3
[33]	New York, U.S	arrivals, departures, and weather information	Long short-term memory (LSTM) model	RMSLE within 0.45 to 0.50,
[35]	New York, U.S	bike usage, weather, time factors, and correlation among the stations and user information	Linear regression, Gradient Boosting Decision and Random Forest	RMSLE around 0.5 for the Random Forest
[36]	Barcelona, Spain	weather, time factors, arrivals and departures	Linear Regression, ARIMA, Prophet, Random Forest	RMSE around 0.4 for Random Forest
[37]	New York, U.S	bike station's demand, type of day (weekday or weekend), and hour	Gradient Boosted Regression Tree (GBRT) and Holtz	MSE around 6 for GBRT and 2 for Holtz
[38]	Washington, U.S.	bike usage, time and weather factors	Ridge, Linear regression, Support Vector Machine, Random Forest, and Gradient Boosted Tree	RMSLE around 0.3 for Random Forest and Gradient Boosting
[39]	Washington, U.S.	bike usage, time and weather factors	Multiple Linear regression model, Random Forest and Gradient Boosting models	Random forest decreased the error by 10%
[40]	Suzhou, China	available bikes, and time factors	Random Forest, Long Short-Time Memory (LSTM) and Gated Recurrent Units (GRU)	Similar behavior of all methods, RMSE around 1 or 2 bikes
[41]	Seoul, Korea	bike-sharing related features, sociodemographic, built environment, weather, land use and landscape factors	Decision Tree (DT), Random Forest (RF), and Extra Tree (ET)	accuracy of 0.71 for DT
[42]	Nanjing, China	bike usage, weather data, air quality and land usage patterns	Long Short-Time Memory (LSTM), Historical Average (HA), Autoregressive integrated moving average (ARIMA), Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), Artificial Neural Network (ANN), and Recurrent Neural Network (RNN)	MAPE between 12.5 and 49.6 for LSTM
[44]	Salamanca, Spain	bike usage, weather and distance information	Random Forest, Gradient Boosting and Extra Tree Regressor models	RMSLE between 0.4 and 0.8 for Random Forest
[45]	Washington, Denver, and Minnesota, U.S.	demographic, built environment, and transportation network factors	Multivariate Regression model	R2 around 0.76

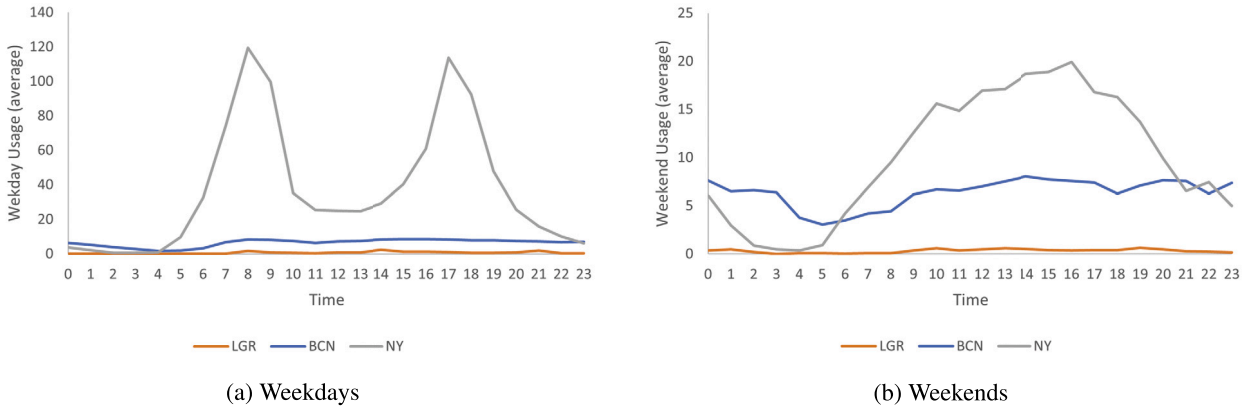
offered by Citi Bike are: single ride (3.99 dollars per trip), day pass (15 dollars per day) and annual membership (185 dollars per year).

Table 2 summarizes the above-mentioned characteristics of the selected BSSs. Weather is also different in these cities. Barcelona has warm weather with soft winter, while New York is colder. However, Logroño has more humidity, especially during autumn and winter. Regarding their usage characteristics, there is a clear difference between them, specially with BiciLog system. It has an average of less than one bike used per hour, while Barcelona or New York BSSs have an average of more than 10 bikes arriving or departing per hour.

Fig. 1 displays the average usage (arrivals + departures) in the most used station in each city. It is evident the usage differences in systems with distinct sizes and maturity. Besides, the usage between weekdays and weekends also varies, especially in such dynamic systems as New York. During weekdays (see Fig. 1-(a)), the average usage in New York most used docking station can reach more than 100 trips (arrivals + departures) during peak hours, while in weekends, they can drop to less than 20. The peak hours also vary when comparing weekdays and weekends. During weekdays, they match business hours. The most popular station in New York

Table 2
BSS characteristics. Three cities with different BSS sizes and characteristics have been selected for this study. Logroño (Spain) called BiciLog considered as a small system has 23 docking stations. Bicing in Barcelona (Spain) with 519 stations is our medium size system and Citi Bike in New York (US) with more than 1 500 stations is a large BSS. Not only the size of the BSS is different, the usage also varies accordingly. The average of arrivals and departures in the small system is less than 1 per hour, while in medium and large systems is more than 10. It shows that bigger BSS are also more dynamic.

Data Sources		Logroño	Barcelona	New York
General	BSS	BiciLog	Bicing	Citi Bike
	Size	Small	Medium	Large
	Inhabitants	150k+	1.6+ million	8+ million
BSS Data	Subscribers	2.9k+	130k+	12+ million
	#bikes	300	7 000	24 500
	#stations	23	519	1 500+
	#arrivals per hour (average)	0.59	11.21	15.71
	#departures per hour (average)	0.59	11.26	15.64
Weather Data	Temperature °C (average)	8.57	12.78	6.91
	Wind Speed km/h (average)	3.11	4.56	2.81
	Humidity %	82.87	74.16	61.26



(a) Weekdays (b) Weekends

Fig. 1. Most used station's usage in each BSS. Citi Bike system shows a more dynamic behavior, especially during weekdays, where the most popular station can reach more than 100 trips (arrivals+departures) at early morning and late afternoon. During the weekend, the trend changes and is reduced to 20-25 trips at peak hours in the afternoon. Similar behavior is presented in Barcelona BSS. However, the quantity of trips is between 10 and 20, in both weekdays and weekends. BiciLog, with less than 3 trips per hour during the week, does not show any particular trend.

is used during the morning, from 7 to 9.00, and in late afternoon from 17 to 19.00. During weekends (see Fig. 1-(b)), this pattern changes and peak hours are moved to the afternoon, from noon to 19.00. In Barcelona, usage remains in same levels, between 10 and 20, for both weekends and weekdays. Moreover, the peak hours during weekdays and weekends are similar to New York. On the other hand, Logroño's most used station has lower levels than the other cities during the week, with an average of less than three trips per hour. BiciLog doesn't show any particular usage trend during the week.

4. Methodology

4.1. Features

The selection of features was based on the information available for the three cities considered, and the knowledge contributed by previous studies [32,40,42] which have already investigated some factors that might affect bike sharing systems' demand. These attributes can be classified as station-related or weather features.

In the first group, there are the following features: Date, Station ID, Day of the week (from Monday to Sunday), Hour interval (4 levels), Total Arrivals, Total Departures. Weather features are: Humidity, Pressure, Temperature, Wind, Weather type (5 levels). Day of the week, time interval, and weather type are categorical variables, while the others are numerical.

4.2. Data collection and cleaning

As mentioned in the previous section, we compared three cities with BSS of different sizes. Therefore, the information needed for this study is available through various sources. To train and evaluate the models, we concentrated on information from October 2019 to February 2020, for the following reasons:

- Users behavior was abnormal during 2020 as a consequence of the COVID-19 pandemic, the lockdown in many cities and several mobility restrictions. So, we did not use data from March 2020 and onwards.
- There is a lack of continuity in Barcelona's information, as the company that provided the bike service in this city was gradually changed during 2019. Therefore, information from the current company is available from August 2019.
- Logroño's BSS data was made available from October 2019. As we are interested in a models' performance analysis during the same period for the three cities, we use October as the starting month of analysis.

Additionally, daily data for the three cities was grouped in 4 time intervals. The main reason for this decision was the low number of pickups and drop-offs in Logroño which could create an hourly database with mostly 0s. Grouping the information in 4 intervals overcome this situation and also helps to solve Logroño's City Hall requirements [46]. The intervals considered are:

- night: from 01:00 to 06:59
- morning: from 07:00 to 12:59
- afternoon: from 13:00 to 18:59
- evening: from 19:00 to 00:59

Data collection process is described as follows:

Weather To have a common source of information, weather data was collected from Time and Date web page. This website reports the historical hourly data for each city for the chosen features of this study. The weather features selected are temperature (Celsius), wind (km/h), humidity (%), atmospheric pressure (mbar), and weather type. Later, weather type feature was classified in five categories: sunny, cloudy, light rain, heavy rain and snow.

Logroño. The information of Logroño's BSS named BiciLog is not publicly available. Therefore, to access and use the BiciLog data, we needed approvals from Logroño's City Hall. Information about real-time trips as well as historical data, which includes arrivals and departures, were made available through a private website maintained by the Instituto Tecnológico de Castilla y León (ITCL). BiciLog data was collected respecting data protection laws (GDPR). This BSS has a total of 23 docking stations distributed across the city, but three of them were opened during the last quarter of 2021. Therefore, only 20 stations were used.

Barcelona. Barcelona BSS called Bicing has data available through the Open Data BCN website. This information is updated monthly. Bicing databases contain more than 3.5 million records for each month, as the information is collected approximately every 5 minutes. Currently, Bicing has 519 docking stations. However, during the period considered, only 409 stations were working.

New York. Similar to Barcelona, New York BSS, Citi Bike, has publicly available data through an open data website. The monthly data uploaded has been already processed to remove trips that correspond to test stations or trips that are below 60 seconds in length. For the period under analysis, data collected corresponds to 840 docking stations that were operating during that period.

Data cleaning. Data processing, cleaning, and preparation procedures were performed in R. Model training and evaluation was also done in R using a Windows 10 machine with an Intel Core i5-6200U CPU, 2.30 GHz, 16 GB RAM.

As each city has its file format, granularity, and number of docking stations, we cleaned the data and standardized the database format to be used in the forecasting process. Data cleaning includes deleting information about docking stations that even when appearing in the dataset are still not used, as well as others that were not working properly or were used as test stations. Moreover, garbled data (around 1% in each city) such as dates outside the analyzed range (from October 2019 to February 2020) or negative bike availability, was removed. For each city, we calculated additional information needed to train the models, such as day of the week, time interval, and the total number of arrivals and departures by station and time period. Missing data, which was less than 1% for the three BSS, was imputed using linear interpolation. Finally, weather and bike usage data were merged and grouped by time intervals for each city. We used weather data from the previous period (t-1) to train and predict bikes' arrivals and departures. For non-time series models (Random Forest, Linear Model, XGB) we also employed the usage information from the last 8 periods (t-1 to t-8) to help the models to learn the patterns for the last periods. For ARIMA and Prophet this was not necessary, these time series models have a seasonal component which we set to 4 (for the 4 daily hour intervals) to make the five models comparable. Table 3 summarized the features used in the models.

After cleaning, the three datasets were split into a training and a testing set. The training set contains data from October 12th 2019 to February 22nd, 2020. The last week of February, from 23rd to 29th, was used to evaluate the performance of the models. To try to replicate what a model with real-time data will do, we used a one-step forecast approach, i.e., we predicted the value for the next day. Then, training and test sets were updated to predict data for the next day. As each station has its own usage behavior patterns, and we aim to predict station-level usage, a prediction model was trained for each station. Likewise, we used departures data to predict bikes that are leaving from a station and arrivals data to forecast bikes arriving to each station.

4.3. Models

As was mentioned in Section 2, several models have been tested by different authors to predict bike sharing demand. Some of them are extensively used in many domains, while others are more oriented to the bike sharing scenario. Five models have been

Table 3
Description of the data used for training and evaluating the models. Weather features from the previous period (t-1) are used in all ML models. For non-time series algorithms (Random Forest, Linear Model and XGBoost), arrivals and departures data from the last 8 periods (t-1 to t-8) was used to help the models learn the usage patterns. For time series models (ARIMA and Prophet) a seasonal component was included in the model.

Category	Attribute	Type
Time	Date	DateTime
	Day (Mon-Sun)	Categorical
	Time Interval (1-4)	Categorical
Weather (t-1)	Pressure	Numerical
	Humidity	Numerical
	Temperature	Numerical
	Wind	Numerical
	Weather Type (1-5)	Categorical
Station Data (t-1 ,..., t-8)	#arrivals	Numerical
	#departures	Numerical

chosen for this study. These models have been previously used in BSS forecasting, and they are currently available in many libraries or packages in R and Python. A brief description, together with the model parameters as well as their optimization process, is presented in this section.

Linear regression model (LM). It is one of the most extensively studied and used forecasting algorithms. It models the relationship between two or more variables by fitting a linear equation to the observed data. One of the greatest advantages of linear models is that their unknown parameters and statistical properties are easier to estimate from the data itself. It can be expressed as:

$$y_i = \beta_0 + \beta_1 X_{i1} + ... + \beta_p X_{ip} + \epsilon_i \quad i = 1, ...n \tag{1}$$

Where: $Y = (y_1, \ldots, y_n)'$ is the endogenous variable (n-dimensional vector), $X = (X_{ij})_{i=1,\ldots,n;j=1,\ldots,p}$ is the matrix of regressors or explanatory variables ($n \times p$ matrix), $\beta = (\beta_0, \ldots, \beta_p)'$ is a (p+1) dimensional vector of the regression coefficients and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)'$ is the error term (n-dimensional vector). This last variable captures all other factors that could influence the endogenous variable and are not included in the explanatory variables. The dependent variable should always be numerical, while the independent variables may be numerical, binary, or categorical.

Many techniques have been developed to estimate the model parameters. The most widely used is the Least square estimation, which tries to minimize the sum of the mean square loss. The techniques to estimate the parameters will not be covered as they are out of the scope of this study. More information can be found in [47].

Autoregressive integrated moving average (ARIMA). It is one of the most common approaches in time series forecasting. It is a combination of the differenced autoregressive model and the moving average model [48]. It is a method of forecasting which uses its lags as predictors, i.e., its predictors are dependent on each other. It is usually denoted as ARIMA(p,d,q) where p is the number of time lags of the autoregressive model, d is the degree of differencing, and q is the order of the moving-average model. All these parameters are non-negative integers. Its components are: *i) Autoregression (AR)* which shows that the time series variable is regressed on its own lagged or past values.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \epsilon_1 \tag{2}$$

Where: Y_t is the predicted value, p is the lag order, $Y_{t-1}, \ldots, Y_{t-p}$ are the lagged values and β are the estimated coefficients and α is the intercept term.

ii) Integrated (I) part represents the differenced values of d (degree of difference) order necessary for the time series to become stationary, it means raw values are replaced by the difference between data and the previous values.

iii) Moving Average (MA) part indicates the dependency between an observation and its residual error, i.e., the forecast error can be represented as a linear combination of past errors.

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q} \tag{3}$$

Where: Y_t is the predicted value, q is the order of the moving average, $\epsilon_{t-1}, \ldots, \epsilon_{t-q}$ are the errors and ϕ are the estimated coefficients and α is the intercept term.

ARIMA general formula can be expressed as follows:

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + ... + \phi_q \epsilon_{t-q} \tag{4}$$

Where: the first part corresponds to the estimation of the AR part and the second to the MA term for the stationary time series, i.e., when the time series was differenced at least once.

The optimization of parameters (p,d,q) for ARIMA, can be performed as follows:

- i) Find the right order of d (degrees of difference) to make the series stationary. ACF plots or the Augmented Dickey Fuller test are commonly used.
- ii) Estimate the coefficients of AR terms and MA terms based on the identification of p and q orders by using the ACF and PACF of the stationary series.

As was mentioned before, we have included a seasonal component in the time series models. In this case, when the seasonality contributes to the forecast, the model is called SARIMA. It adds three new parameters (P, D, Q) to specify the autoregression (AR), differencing (I) and moving average (MA) for the seasonal component of the series. Additionally, a m parameter for the period of the seasonality is also added. Therefore, it can be noted as:

$$SARIMA(pdq)(PDQ)_m \tag{5}$$

As in the case of ARIMA, the values (P, D, Q) for the seasonal part of SARIMA can be obtained from the ACF and PACF plots [49]. Many current packages already implement the analysis of these plots and tests, so SARIMA parameters can be easily estimated.

Random forest (RF). It is a supervised learning algorithm with an ensemble learning method that consists of randomly generating subsets of the features to build smaller trees and train them using the bagging method. It is widely used in Regression and Classification problems. For each tree, the output is the predicted number (regression) or class (classification). The final output of the Random Forest algorithm is the average prediction in the case of regression problems, or the class selected by most of the trees for classification problems.

Random Forest tackles the overfitting problem that individual decision trees usually suffer by training decision trees with different parts of the training set. It creates random subsets of the features, builds smaller trees using those subsets and combines them, reducing the variance. Thanks to the combination of many learning models, Random forest results always outperform any individual tree model. However, for numerous trees, the algorithm can be slow and require plenty of computational resources. More information can be found at [50].

Prophet. It is an open-source software implemented in R and Python that was released by Facebook in 2017 [51]. It was designed as a procedure for time series prediction based on an additive model where non-linear trends are fitted with different seasonality (yearly, weekly, and daily), and also considering holiday effects. Therefore, it has a better performance for time series with strong seasonality and enough historical data. Moreover, it is robust to outliers, missing data, and dramatic trend shifts. It can be formulated as follows:

$$Y_t = g_t + s_t + h_t + \epsilon_t \tag{6}$$

Where g_t is the trend function that model non-periodic changes, s_t models periodic changes, h_t represents holidays effects and ϵ_t is the error term.

The g_t growth function models the trend of the data. What differentiates Prophet from other tools is that it considers that trends can be present at all points in the data and can be altered using change points. Change points are those moments where the data shifts direction. Prophet detects change points automatically or gives the option to users to set them by themselves. Holidays are usually based on US holidays, but users can use their dates too. Holidays add or subtract value from the prediction from the growth and seasonality terms based on historical data (past holidays). What makes Prophet so popular nowadays is its ease of implementation, and that is intuitive to non-experts.

Extreme gradient boosting (XGB). It is an optimized library based on Gradient Boosting (GB) framework, and designed for speed and performance, i.e., it builds trees in parallel instead of sequentially like in Gradient Boosting. Similar to Random Forest, XGB builds a model based on multiple decision trees. However, the main difference between them is that Random Forest uses a "bagging" method that minimizes the variance and overfitting, while the Gradient boosting method minimizes the bias and underfitting.

XGB is built upon *i)* supervised machine learning: using labeled datasets, *ii)* decision trees, *iii)* ensemble learning: by combining multiple algorithms to obtain a better performance, and *iv)* gradient boost: with each iteration, XGB uses the error residuals of the previous model to improve the next one. The final prediction is the weighted sum of all trees' predictions. More information is available in [52].

Given the characteristics of the selected methods, we can classify them in two groups: time series models (ARIMA and Prophet) and non-time series models (Linear Model, Random Forest, XGB). The time series models are expected to have a better performance when working with BSS data, as these models are prepared to identify seasonal patterns. In this study, the non-time series models were trained using the arrivals and departures from the last 8 periods to help the models to learn BSS data patterns. Moreover, traditional methods such as the Linear Model, ARIMA, or Prophet, allow users the selection of features and the configuration of parameters, which can also help to improve the model performance. Nevertheless, parameter configuration can be complex and time-consuming for some models like ARIMA. On the other hand, machine learning techniques (Random Forest and XGB) can easily

and automatically identify patterns that are difficult for humans to discover and configure through model parameters. In addition, machine learning forecasting results tend to be more accurate given an initial set of hyperparameters. However, machine learning techniques usually require more data to train and more computational resources. In the case of this study, the hyperparameters for the machine learning methods have been set previously and the auto-arima function has been used, to facilitate the analysis of the scalability of the prediction methods for the three proposed scenarios. In the case of overfitting, which is an important limitation that should be considering in forecasting techniques, we have tested each model using in-sample and out-of sample method. The in-sample method refers to how well the model fits the training data. While, out-of-sample method is used to evaluate the model with unseen data which, in this study, has been saved in the test dataset. In our case, in-sample and out-of-sample results were of a similar order, which is a good indicator of the models' performance, as observations out of the training set were predicted with a similar accuracy as observations in the training set.

4.4. Error metrics

Once we have calculated an estimation, we need a way to determine its accuracy. In the literature, many error metrics have been proposed. Here, we only visit the subset of the most popular ones that will be used later in the evaluation.

Root mean square error (RMSE) It is a measure of the differences between the values predicted by a model and the real observed values. RMSE is the square root of the average of squared errors, formally defined as follows:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{7}$$

Where $\hat{y}_i$ are the predicted values, y_i are the observed values, and n is the number of observations.

For general interpretation, a lower RMSE is better than a higher one. Nevertheless, these error measures depend on the scale of the data used for the forecasting tasks. This means comparisons across different types or scales of data would be invalid. RMSE is also sensitive to outliers, as the effect of each error is proportional to the size of the squared error.

Normalized root mean square error (nRMSE) It facilitates the comparison between models or data with different scales. There are several methods to normalize the RMSE. From the candidates, we will use the mean, as we are comparing models based with the same dependent variable and with similar data treatment. nRMSE is calculated as:

$$nRMSE = \frac{RMSE}{\bar{y}} \tag{8}$$

Where $\bar{y}$ is the mean of the observed values. The main difference with RMSE is that nRMSE does not have units. Therefore, it could be interpreted as a relative measure.

Root mean square logarithmic error (RMSLE) Nowadays, it is one of the widely used error metrics for bike sharing demand and especially in many Kaggle competitions due to its robustness to outliers. It is defined as:

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(\hat{y}_i + 1) - log(y_i + 1))^2} \tag{9}$$

Where $\hat{y}_i$ are the predicted values, y_i are the observed values, and n is the number of observations.

The use of the logarithm ensures that outliers are drastically scaled down, almost nullifying their effect. In our study, it means that errors produced during peak time intervals do not dominate the one produced during non-peak intervals. Considering the properties of logarithms, RMSLE can be interpreted as the relative error between the predicted and the actual values. Additionally, RMSLE allows the comparison of data with different scales, as this error metric is relative and does not have units.

5. Results

5.1. Overall results

The error metrics for both arrivals and departures are summarized in Tables 4, 5 and 6 for Logroño, Barcelona, and NYC, respectively. As it was expected, the precision of the prediction is different in each city. Moreover, not a single model outperforms all the others in terms of forecasting performance in the three considered scenarios. Random Forest (RF) achieved the lowest error metrics when predicting, both arrivals and departures, in Barcelona. Bicing errors (see Table 5) reached a RMSLE of 0.36 when predicting arrivals using the test set, while for departures the RMSLE was 0.37. For the other cities, the Prophet algorithm achieved better scores. In the case of Logroño, the RMSLE metric (see Table 4) has a value of 0.41 for both, arrivals and departures. In Citi Bike system, this error metrics equal to 0.61 (see Table 6).

A close look at the Tables 4, 5 and 6, gives also interesting insights when comparing the models' performance in each city. For example, error metrics are pretty close using Logroño system data if we compare all models used. More significant differences appear for Barcelona and New York data when comparing the error metrics between the prediction models trained. Additionally, the results

Table 4
Logroño - BiciLog Results. Prophet outperforms other algorithm's performance in both arrivals and departures. However, the values of error metrics are quite close for all models trained.

	Arrivals				Departures			
	RMSE	nRMSE	RMSLE	Training Time (min)	RMSE	nRMSE	RMSLE	Training Time (min)
ARIMA	0.908	1.311	0.413	10.0	0.926	1.329	0.416	4.8
Linear Model	0.959	1.385	0.440	0.1	0.999	1.434	0.448	0.1
Prophet	**0.886**	**1.280**	**0.410**	**3.2**	**0.906**	**1.300**	**0.410**	**3.1**
Random Forest	0.935	1.350	0.423	3.0	0.985	1.414	0.435	3.1
XGBoost	0.986	1.425	0.477	12.4	1.01	1.450	0.473	13.4

Table 5
Barcelona - Bicing Results. Random forest gives the best results when predicting arrivals and departures. Similar as in Logroño and New York, ARIMA and XGBoost take longer times to be trained. The differences in error metrics between the models considered are becoming more evident with a bigger data size and a more dynamic system.

	Arrivals				Departures			
	RMSE	nRMSE	RMSLE	Training Time (min)	RMSE	nRMSE	RMSLE	Training Time (min)
ARIMA	3.824	0.318	0.380	299.3	3.871	0.319	0.379	121.3
Linear Model	4.441	0.369	0.426	1.9	4.360	0.359	0.422	2.2
Prophet	3.856	0.321	0.376	59.1	3.849	0.317	0.378	59.1
Random Forest	**3.641**	**0.303**	**0.367**	**72.8**	**3.762**	**0.310**	**0.375**	**73.2**
XGBoost	4.174	0.347	0.436	275.4	4.224	0.348	0.437	274.9

Table 6
New York - Citi Bike Results. Both, arrivals and departures are being predicted more accurately by Prophet. ARIMA, Random Forest and Prophet have close error metrics and similar performance. However, ARIMA needs longer time to be trained.

	Arrivals				Departures			
	RMSE	nRMSE	RMSLE	Training Time (min)	RMSE	nRMSE	RMSLE	Training Time (min)
ARIMA	9.219	0.612	0.621	572.1	8.794	0.587	0.619	214.8
Lineal Model	10.846	0.721	0.661	4.3	10.275	0.686	0.662	4.1
Prophet	**9.225**	**0.613**	**0.619**	**120.7**	**8.773**	**0.586**	**0.618**	**121.0**
Random Forest	9.131	0.607	0.633	164.8	8.960	0.598	0.636	167.0
XGBoost	10.356	0.688	0.748	581.6	10.476	0.699	0.749	582.8

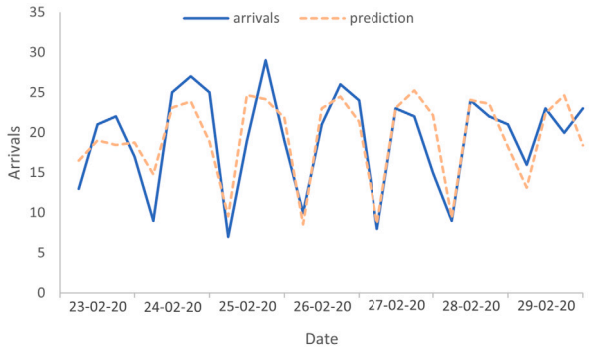
show that ARIMA model has also a high performance, and it is the second-best model in all cities. ARIMA error metrics are pretty similar to Prophet metrics for Logroño and New York and to Random Forest in Barcelona BSS.

Even when the trained models usually follow the trends well, the difference of performance between cities is quite evident. In the case of Logroño, with a small BSS, the difference between the real and predicted data is less than 1 bike. However, in such a small system, this could represent a big percentage error (nRMSE equal to 1.2 for arrivals and 1.3 for departures) especially in those stations with few or none movements during certain time intervals of the day. This situation becomes more visible if we compare the results of New York City. In this case, the difference between the real and predicted value is close to 9 bikes for arrivals and departures. Despite this, the nRMSE has a value of 0.6, half of the error for Logroño BSS.

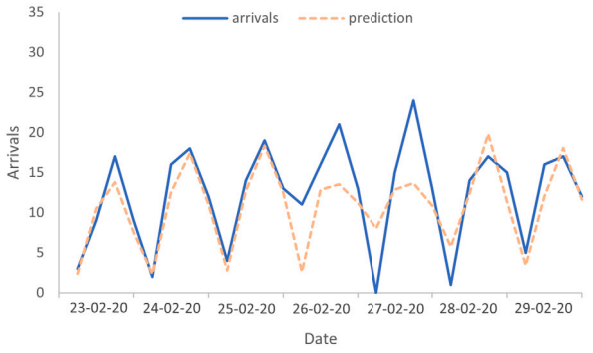
Based on the comparison of the results for the three cities, we can conclude that performance is closely related to the dynamic of the BSS. The more dynamic the system is, the better is the model accuracy. We can also check that prediction algorithms have learned better the trends in data for Barcelona BSS. Fig. 2 displays the real value of arrivals and the prediction using Random Forest, which is the model with the best performance in this BSS. On the left side, the Fig. 2-(a) contains the data for the most used station. The predicted value follows pretty well the trend of the arrivals, especially when there are peaks of usage. On the right side, the Fig. 2-(b) shows the data of the station with the highest error metrics using the Random Forest model. In this case, RF failed to predict the arrivals when the trend suddenly changes and the usage is drastically increased or decreased.

The case of Logroño BSS is different. This BSS has a less dynamic usage behavior, which makes models fail to learn their patterns. Fig. 3 shows the real and estimated arrivals values in two bike stations of BiciLog system using Prophet. On the left, in Fig. 3-(a), the more dynamic station is displayed and readers can see that Prophet is failing to capture some peaks of usage. On the right side, Fig. 3-(b) contains the arrivals and estimated values for the station with the highest RMSLE, it means the station where Prophet had more failures to predict the arrivals. For this station, it is evident that our prediction model has learned a pattern with a lower number of arrivals, and it is not able to estimate peaks of usage.

Finally, similar plots are displayed for CitiBike system in Fig. 4. New York best prediction model, Prophet, have learned quite well the trends in the data, especially for the most used station, as it is depicted in Fig. 4-(a). Even when there are some usage peaks on the arrivals that Prophet is underestimating, the general trend for this station is well captured by this model. However, in the station with the highest error metrics (Fig. 4-(b)) Prophet is underestimating the arrivals. Actually, we can see that towards the end of the

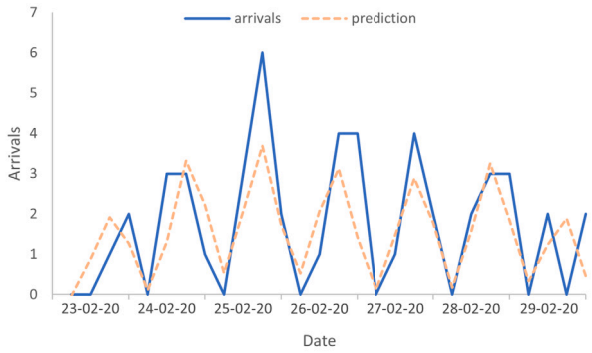


(a) Most Used station: RF predicted values vs real ones

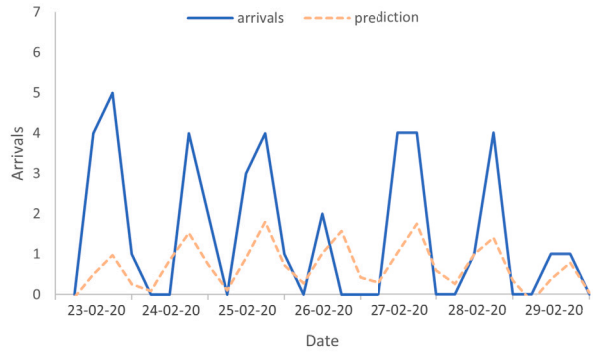


(b) Highest RMSLE station: RF predicted values vs real ones

Fig. 2. Barcelona: Estimated and real arrivals values for the most used (a) and for the station with the highest RMSLE (b) using Random Forest. The left view depicts a more constant trend which is accurately predicted by RF algorithm. On the contrary, the right view shows that RF is failing to predict changing patterns in the arrivals flow.

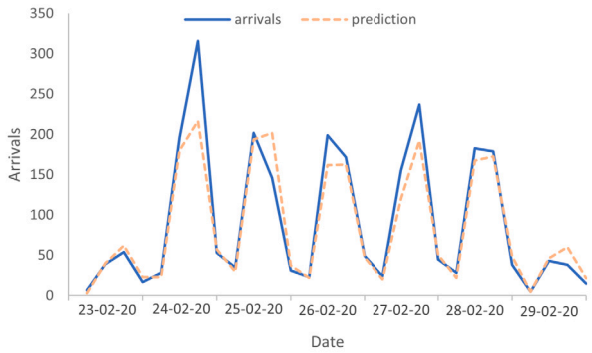


(a) Most Used

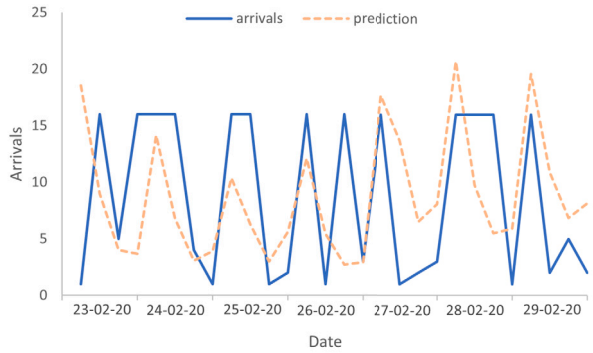


(b) Highest RMSLE

Fig. 3. Logroño: Estimated and real arrivals values for the most used (a) and for the station with the highest RMSLE (b) using Prophet. The left side view shows that arrivals in the most used station are being predicted more accurately. The right view displays that forecast models are predicting less than two bikes arriving each interval. However, the real value of arrivals is bigger than two in most of the periods.



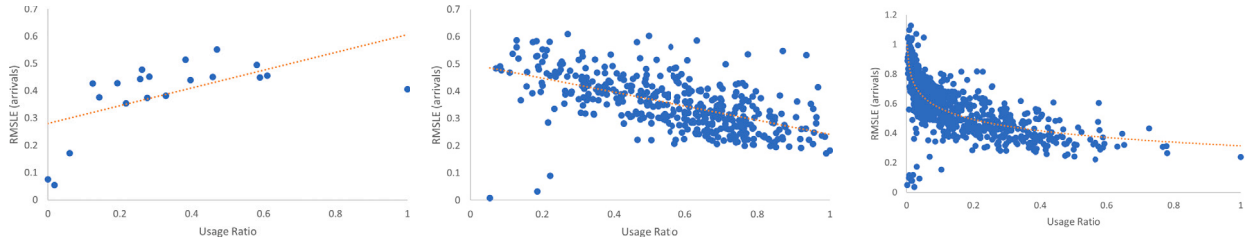
(a) Most Used



(b) Highest RMSLE

Fig. 4. New York: Estimated and real arrivals values for the most used (a) and for the station with the highest RMSLE (b) using Prophet. The left view shows the prediction algorithm has learned the trends properly and can estimate the arrivals under certain ranges. On the opposite, the right side view displays that Prophet is failing to estimate the arrivals, especially during intervals with few or non movements.

data series, the arrivals' prediction has an increasing trend while the actual values have a more stable pattern that never exceeds 15 arrivals.



(a) Logroño - Bicilog (b) Barcelona - Bicing (c) New York - Citi Bike

Fig. 5. Error metric vs. Usage ratio. Logroño shows a positive relationship between the error and the usage ratio, the more used is the station, the bigger the error is. This relationship is negative for Barcelona and New York, i.e., the error is lower for those stations with higher usage ratios.

5.2. Impact of usage ratio

To get a better understanding of the model selected, Random Forest for Barcelona, and Prophet for Logroño and New York City, we analyzed the relationship between the model precision and the station usage ratio. We have created this metric and defined it as the sum of the number of trips (pickups and drop-offs) by time interval. For a better interpretation, we re-scaled the ratio to make all the elements lie between 0 and 1, having a common scale. Formally, it can be defined as:

$$use_{ijk} = \sum arrivals_{ijk} + \sum departures_{ijk}$$

$$UsageRatio_{ijk} = \frac{(use_{ijk} - MinUse_{jk})}{MaxUse_{jk} - MinUse_{jk}}$$

Where i = *StationID,* j = *day,* k = *hour interval*

In each city, the usage ratio and the error metrics were calculated for each docking station. The results are displayed in Fig. 5. The relationship between these metrics is different in each city. In the case of Logroño (see Fig. 5-(a)), the error metric has a positive and linear relationship. When the station usage ratio increases, the error metric is larger. On the contrary, the relationship is negative but also linear for Barcelona (see Fig. 5-(b)), as the error metric decreases when the usage ratio increases. The plot for New York (see Fig. 5-(c)) shows a strong and logarithmic relationship between the error and the usage ratio. However, it is evident that most of the docking stations with higher errors are those with lowest usage ratio.

5.3. Error analysis by temporal distribution

Similarly, error metrics were computed for each station, time interval and dates in the test set. It means in the last week of February 2020. Fig. 6 displays a heatmap for each city, which shows the error in each time interval by docking station. Stations are ordered by their usage ratio. As was shown in section 5.2, the relationship between the error metric and the usage ratio is positive for Logroño (see Fig. 6-(a)) and negative for Barcelona (see Fig. 6-(b)) and New York (see Fig. 6-(c)). We can also see interesting patterns in Barcelona and New York. For example, the error is higher for the first interval (from 01:00 to 06:59) of each day. In the case of Logroño there is no similar pattern and the error seems not to be related with the time intervals.

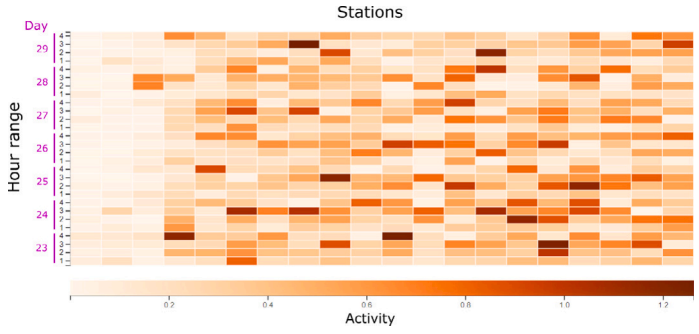
5.4. Error analysis by spatial distribution

Finally, we computed the error metric in each docking station and plotted them in a map to investigate common patterns. Fig. 7, 8, and 9 illustrate these findings. As was commented before, our three selected cities show different patterns according to their BSS size. Logroño, considered in this investigation as a small BSS, exhibits bigger prediction error (see Fig. 7-(a)) in those stations which have higher usage ratio (see Fig. 7-(b)). In this case, the most used stations are close to the city center. On the contrary, Barcelona with a medium BSS and New York with a large BSS, display lower prediction error in those stations located in the city center (see Fig. 8)-(a) for Barcelona and Fig. 9-(a) for New York), which are also the most used stations (see Fig. 8)-(b) and Fig. 9-(b) respectively).

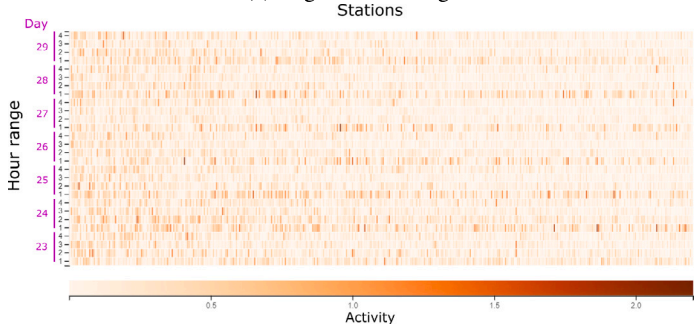
5.5. Training time

With the aim to compare training time for different data sizes, models for all cities were trained using the same hyperparameters. Tables 4, 5, and 6 show the training time in minutes for both arrivals and departures forecasting. We can clearly see that training times increases when the data sizes are larger.

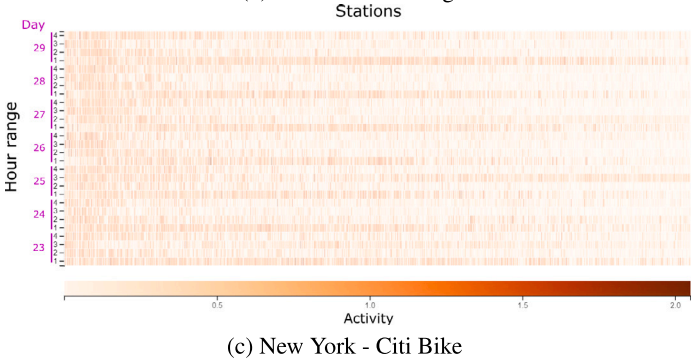
In all cities, training a Linear Model is time cost-efficient. For XGB model, we limited the number of rounds to 100 and test a depth of trees of 10 and 15. Therefore, it takes longer times to train. Similarly, training ARIMA models is taking longer times as we are using the *auto.arima* function in R and feeding time series with a frequency equal to 4 (due to the four daily time intervals). We used the default parameters in R for Random Forest and Prophet to train the models for the three cities. In the case of Logroño, the training time is similar for both algorithms. This similarity disappears, and the differences are more evident when the data size



(a) Logroño - Bicilog

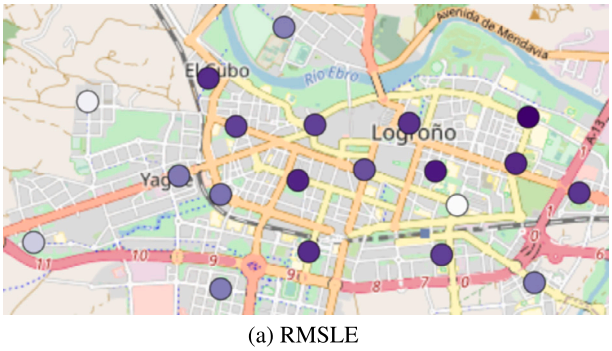


(b) Barcelona - Bicing

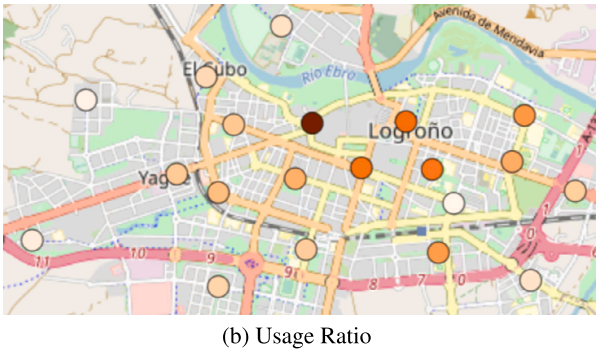


(c) New York - Citi Bike

Fig. 6. Detailed analysis of the error metric. The vertical axis shows the days (days 23-29 in purple) subdivided in time intervals along the day (1-4 in black). New York and Barcelona show a clear pattern: during the first interval (01:00 to 06:59) the error is higher. However, in Logroño, no clear patterns are visible.



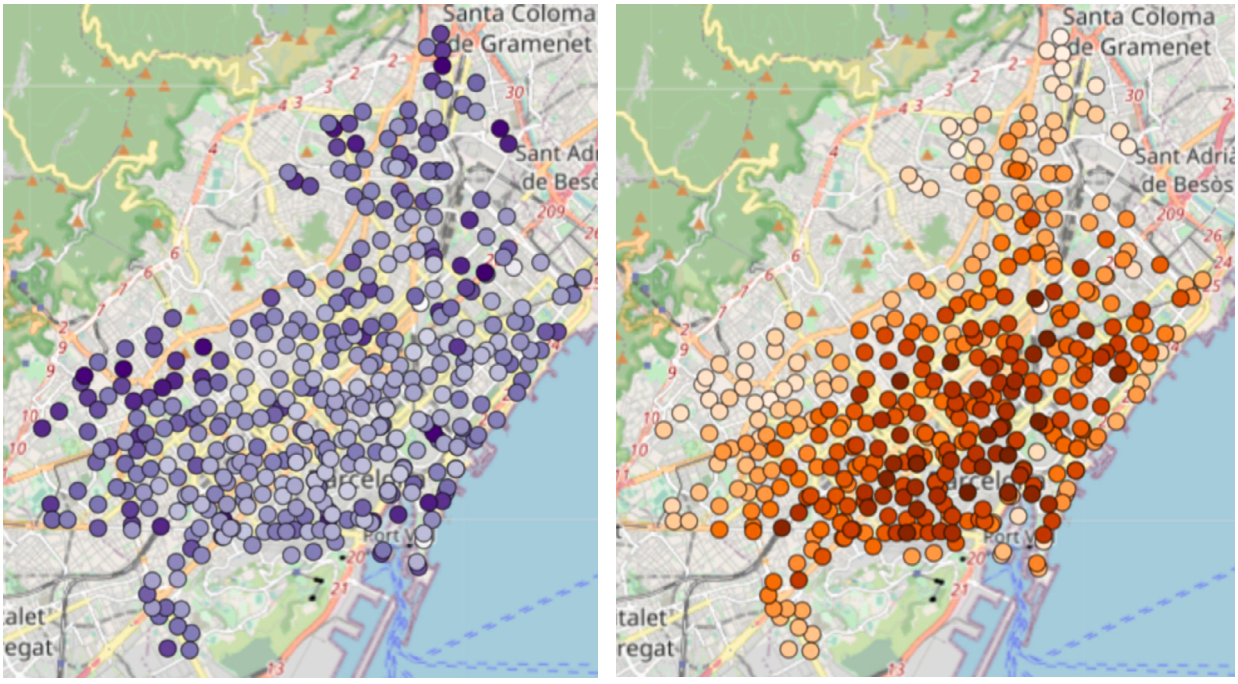
(a) RMSLE



(b) Usage Ratio

Fig. 7. Error metrics distribution and usage ratio for Logroño. The most used stations are close to the city center and have higher RMSLE.

becomes larger. Tables 5, and 6 show how Random Forest takes longer time than Prophet to train in Barcelona and New York data, i.e., datasets with larger sizes.



(a) RMSLE (b) Usage Ratio

Fig. 8. Spatial distribution of error metrics and usage ratio in Barcelona. Similarly to Logroño, stations with higher usage ratios are close to the city center. However, the relationship is opposite, and these stations have lower RMSLE.

6. Discussion

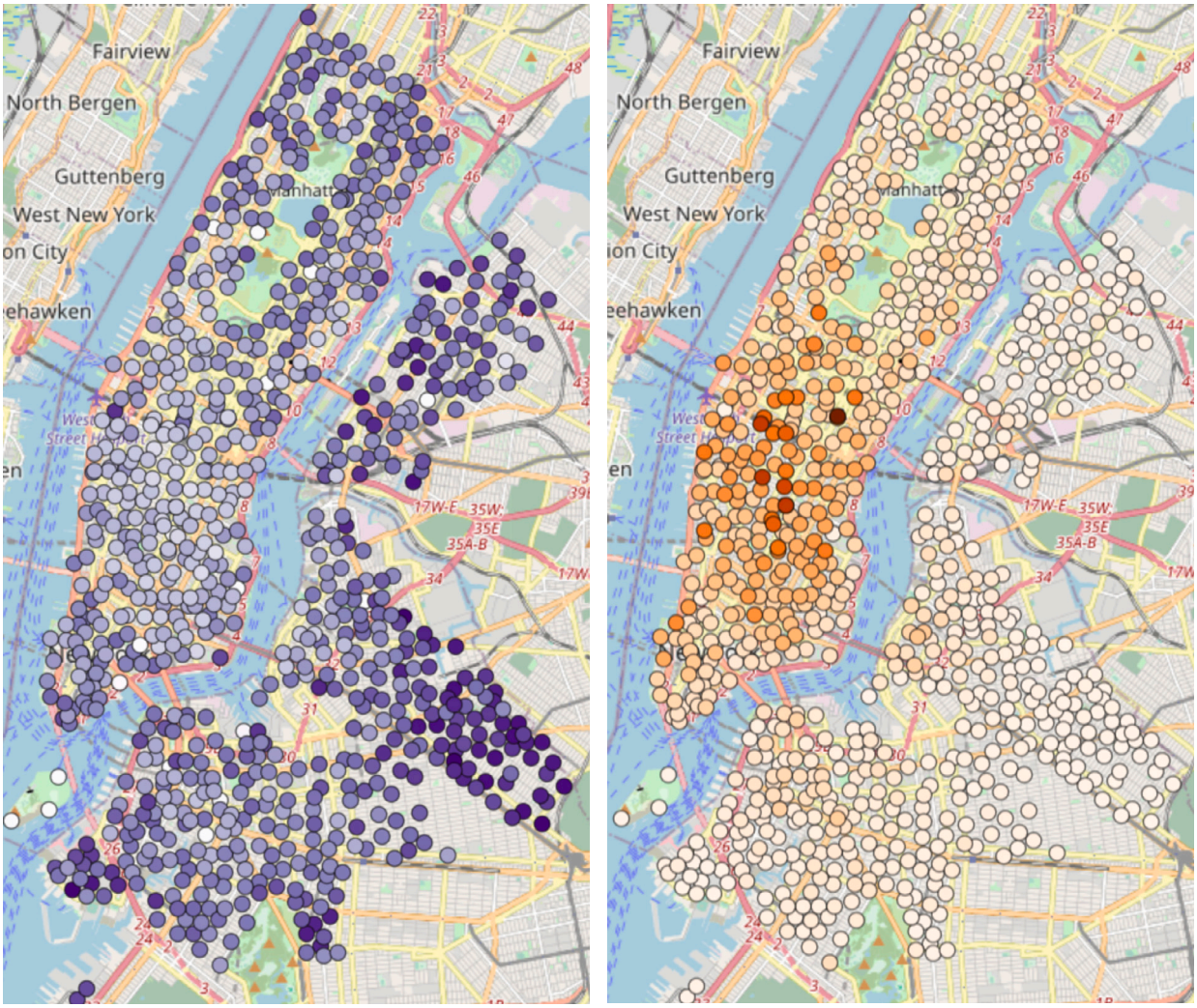
During data cleaning, some garbled data was removed from the databases. For the three cities considered, this represents less than 1% of the total size of the dataset and removing it did not generate any bias, as there is not enough data to perform a long-term analysis. Similarly, for all databases, imputation of missing information was also performed, and it represents less than 1% of the total set. It is considered that less than 5% of missing data [53] will not have any consequence on the quality of statistical inferences.

With the aim of comparing results under the same conditions, we have used the same features in the dataset and same time intervals (From Oct. to Feb.) which cannot be optimal as they do not capture each BSS usage activity. To improve the results, it will be necessary to analyze the specific conditions of each city and gain understanding of the features which have more impact on bike sharing demand. Besides, the periods needed to train the data and capture the trends can vary in each city. For instance, Logroño or New York with a strong weather variation during seasons will need at least one year of data to learn data patterns. While other cities with less variation in the weather will need fewer data to train the models. Similarly, the four time intervals used to train and evaluate the models (see Section 4.2) can be optimal for Logroño, but for bigger and dynamic systems such as Bicing or Citi Bike, it will be better to use more granularity in the data.

Different from other studies that only uses a single metric to evaluate model performance [44,45,38], during our investigation, we have also highlighted how important it is to analyze multiple error metrics and not rely on a single one. This is especially important in our study as we compare both, different models and different datasets. Each metric considered can show a different perspective to be analyzed, which helps to increase the quality of the models' results interpretation. For example, considering Tables 4, 5, and 6 and checking the RSME metric, one can assume that predictions are more accurate in Logroño, as the values of RSME are lower than 1. This means that models incorrectly predict less than 1 bike when comparing with the real value. However, the percentage error (nRMSE) shows that any of the models has accurately estimated the trends in Logroño BSS and that actually an error of 1 bike can represent more than 100% in a system with relatively low usage (close to zero in some time intervals). This is not the case for Barcelona or New York City. In the case of Barcelona, the RSME shows around 4 bikes error when predicting arrivals or departures. Nevertheless, in a system with such a dynamic usage and size, this could represent a deviation of only 30% to the real value.

Additionally, models performance seems to be more similar in Logroño, as the error metrics have minimal variations. However, these differences become more evident when the data size increases and the system is more dynamic. Even when we are analyzing the same features, Barcelona and New York data have a different scale, if we consider the usage behavior of their BSS. As it is shown in Table 2, the average of arrivals and departures in Barcelona is more than 10 times the value in Logroño, and for New York it reached more than 15 times. This makes that differences in models' performance metrics become more evident, and again emphasizes the importance of choosing the adequate metrics to compare the prediction performance when the data scale is different.

Our results show a low performance for the Linear Model, as was also found by previous studies [35,39]. In fact, this algorithm is usually used as a baseline in many studies. On the other hand, ARIMA model has showed a good performance. The results displayed



(a) RMSLE (b) Usage Ratio

Fig. 9. Spatial distribution of error metrics and usage ratio for the case of New York. Similar to Barcelona BSS, the stations with higher usage ratios have lower RMSLE.

in Tables 4, 5, and 6 show that ARIMA is the second-best model in all systems. The error metrics are very close to the models with the best performance (Prophet in Logroño and New York, and Random Forest in Barcelona). However, the training times are larger for ARIMA as we are using the *auto.arima* implementation in R where several models parameters are tested. Additionally, we have used a SARIMA model with a seasonal pattern equal to 4, considering the four time interval periods that we have for each day.

As we can see in Fig. 2, Random Forest has learned the patterns in Barcelona arrivals and departures quite well. Random Forest has also been selected as the best algorithm in previous studies [44,43,35,38] and, similar to us, these authors also found Random Forest time cost-efficient. In the case of New York, usage trends have been learned by Prophet and the prediction is accurate for those stations with higher usage ratios, as it is displayed in Fig. 4-(a). However, Prophet fails to make an accurate prediction in stations where there are continuous time intervals with few or none trips (see Fig. 4-(b)). For New York, the previous studies that compare the performance of different prediction models found that Random Forest outperforms other models like Gradient Boosting [43] or LSTM [42,40]. However, none of these studies have tested Prophet algorithm and the data used differs to the one considered in this paper due to the data granularity and the number of docking stations under analysis. Prophet also outperforms other models in Logroño. In this case, those stations with higher usage ratios are also the ones with higher error metrics, and models are not learning the trends properly (see Fig. 3). Few studies have been done for smaller BSS. For example, Lozano [44] used data from Salamanca (Spain) and found that Random Forest gave better results compared with other prediction models. However, the data period considered is longer than ours, and the granularity is also larger, which makes not possible comparisons between these BSS results.

The error metrics vary in each city, and it becomes evident that models have to be fine-tuned according to the characteristics of each dataset to improve the accuracy of the prediction. Nevertheless, without fine-tuning the prediction models the results for Barcelona using Random Forest are similar to the results found by previous studies [38,33], and in Kaggle competitions. In addition,

the fact that the data used has been divided in four daily time intervals due to the low usage in the BiciLog system (see Section 4.2) can affect the accuracy of the predictions. To avoid this, some authors have reduced the data granularity (for example, predicting by day) [13], or considered high demanded hours and high demanded stations [32], or clustering stations [31]. In our case, the analysis has been centered on the scalability of the prediction algorithms and model fine-tuning in each city is part of the future work.

7. Conclusions and future work

In this paper, we have set the bases to evaluate how well different prediction algorithms, which have been extensively used in previous literature [35,38–40,42–45], can adapt and perform when the conditions of the dataset vary. Given that previous studies have not focused on the scalability of prediction models in different BSS, we covered this gap in the literature by predicting bikes arrivals and departures in three BSS with diverse usage behavior and size characteristics, and using a set of methods that cover traditional, linear, time series and machine learning algorithms. We have compared the performance of five models to predict short-term station level arrivals and departures in different Bike Sharing Systems (BSS). We used data from BSS of three cities with different sizes and characteristics, and classified them as (i) small, located in Logroño (Spain) with 23 docking stations, (ii) medium, in Barcelona (Spain) with more than 500 stations, and (iii) large, in New York City with more than 1500 docking stations. The algorithms used, ARIMA, Linear Regression, Random Forest, Prophet and XGBoost, were trained using data from October 2019 to February 2020 due to different data availability and to avoid unusual behavior as consequence of the COVID-19 pandemic and its mobility restrictions. Trips and weather information were part of the dataset used to train and test the algorithms. Model evaluation was done using the last week of February as a test set and RMSE, nRMSE, and RMSLE as error metrics.

Results show that Random Forest had a better performance to predict arrivals and departures in our medium size system (Barcelona). Prophet has outperformed the other algorithms in the small BSS (Logroño) and in New York BSS. Other gaps covered by this study are the analysis of the relationship between usage ratio and error metrics, and the variation of the error metric in each predicted interval. In this context, we found a clear relationship between usage ratio and the error metric. In our small BSS, they have a positive relationship; i.e., the higher the usage ratio, the higher the error. While in medium and large systems, this relationship is negative, and the error decreases when the usage ratio increases. Moreover, model accuracy in the medium and large BSS are directly related with the usage dynamic. For instance, during the first time interval with the lowest usage dynamic the error metrics are higher, and they decrease when the usage dynamic increase in the next time intervals. In fact, other authors have only used time periods [32] or stations [31] with larger usage behaviors. Finally, we demonstrated that the same prediction algorithm cannot be used in BSS with different sizes and usage characteristics, and models need to be fine-tuned, according to these characteristics, to achieve better results. Notice that it is also important a correct analysis and interpretation of the data, as well as choosing a correct error metric to evaluate and interpret model performance between different BSS.

One of the limitations of this study is that even when we considered a number of relevant variables, we did not include other factors that could affect BSS demand, as the information needed was not available for all the systems considered. For future work, we would like to evaluate different features such as the number of schools, different transport means or number of shops centers close to a docking station. These spatial features can have some influence in BSS demand, as it is suggested by Rixey study [45]. Other prediction methods such as Recurrent Neural Networks can also be tested, however they will need a larger data period to be properly trained. Similarly, a wider evaluation with a larger data set and a different time interval length would also be beneficial to understand long-term patterns in the data, and the influence of different seasons (winter, summer, spring, autumn) in the BSS demand. These evaluations will be possible once enough data is available. Another line of investigation we are considering is to extend this analysis to other BSS and study how the prediction models would scale in new BSS. Finally, a real-time testing of the models would be possible if permanent access to the original sources and the possibility of setting up a server to handle data is granted.

CRediT authorship contribution statement

Alexandra Cortez-Ordoñez: conceived and designed the experiments; performed the experiments; analyzed and interpreted the data; contributed reagents, materials, analysis tools or data; wrote the paper.

Pere-Pau Vázquez: conceived and designed the experiments; analyzed and interpreted the data; contributed reagents, materials, analysis tools or data; wrote the paper.

José Antonio Sanchez-Espigares: conceived and designed the experiments; analyzed and interpreted the data; contributed reagents, materials, analysis tools or data; wrote the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

The authors do not have permission to share data.

Acknowledgements

Partially supported by PID2021-122136OB-C21 by Ministerio de Ciencia e Innovación/AEI: (10.13039/501100011033/FEDER, UE), and 2021SGR00613-(ADBD) by Generalitat de Cataluña. The authors would like to thank Logroño City Hall for providing the access to the data.

References

[1] R. Meddin, P. DeMaio, O. O'Brien, R. Rabello, C. Yu, R. Gupta, et al., The Meddin bike-sharing world map, 2020. (Accessed 19 October 2021).
[2] D. Fuller, L. Gauvin, Y. Kestens, M. Daniel, M. Fournier, P. Morency, et al., Use of a new public bicycle share program in Montreal, Canada, Am. J. Prev. Med. 41 (1) (2011) 80–83, https://doi.org/10.1016/j.amepre.2011.03.002.
[3] J. Woodcock, M. Tainio, J. Cheshire, O. O'Brien, A. Goodman, Health effects of the London bicycle sharing system: health impact modelling study, BMJ (2014) 348, https://doi.org/10.1136/bmj.g425.
[4] M. Ricci, Bike sharing: a review of evidence on impacts and processes of implementation and operation, Res. Transp. Bus. Manag. 15 (2015) 28–38, https://doi.org/10.1016/j.rtbm.2015.03.003.
[5] P. Midgley, Bicycle-sharing schemes: enhancing sustainable mobility in urban areas, 8, United Nations, Department of Economic and Social Affairs, 2011, pp. 1–12.
[6] G.R. Raidl, B. Hu, M. Rainer-Harbach, P. Papazek, Balancing bicycle sharing systems: improving a VNS by efficiently determining optimal loading operations, in: International Workshop on Hybrid Metaheuristics, Springer, 2013, pp. 130–143.
[7] J. Shu, M.C. Chou, Q. Liu, C.P. Teo, I.L. Wang, Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems, Oper. Res. 61 (6) (2013) 1346–1359, https://doi.org/10.1287/opre.2013.1215.
[8] X.F. Xie, Z. Wang, Combining physical and participatory sensing in urban mobility networks, https://doi.org/10.13140/RG.2.1.4349.4887, 2014.
[9] A. Cortez, P.P. Vázquez, Advanced visual interaction with public bicycle sharing systems, in: Václav Skala-UNION Agency, 2021, pp. 207–2016.
[10] C.R. Bhat, S. Astroza, A.S. Hamdi, A spatial generalized ordered-response model with skew normal kernel error terms with an application to bicycling frequency, Transp. Res., Part B, Methodol. 95 (2017) 126–148, https://doi.org/10.1016/j.trb.2016.10.014, http://www.sciencedirect.com/science/article/pii/S091926151630460X.
[11] X.F. Xie, Z. Wang, Examining travel patterns and characteristics in a bikesharing network and implications for data-driven decision supports: case study in the Washington DC area, J. Transp. Geogr. 71 (2018) 84–102, https://doi.org/10.1016/j.jtrangeo.2018.07.010.
[12] R. Talavera-Garcia, G. Romanillos, D. Arias Molinares, Examining spatio-temporal mobility patterns of bike-sharing systems: the case of BiciMAD (Madrid), J. Maps (2021) 17, https://doi.org/10.1080/17445647.2020.1866697.
[13] A. Faghih-Imani, N. Eluru, A.M. El-Geneidy, M. Rabbat, U. Haq, How land-use and urban form impact bicycle flows: evidence from the bicycle-sharing system (BIXI) in Montreal, J. Transp. Geogr. 41 (2014) 306–314, https://doi.org/10.1016/j.jtrangeo.2014.01.013.
[14] A. Faghih-Imani, N. Eluru, Analysing bicycle-sharing system user destination choice preferences: Chicago's Divvy system, J. Transp. Geogr. 44 (2015) 53–64, https://doi.org/10.1016/j.jtrangeo.2015.03.005.
[15] I. Kim, K. Pelechrinis, A.J. Lee, The anatomy of the daily usage of bike sharing systems: elevation, distance and seasonality, in: ACM SIGKDD Workshop on Urban Computing, 2020, https://par.nsf.gov/biblio/10205854.
[16] I. Frade, A. Ribeiro, Bicycle sharing systems demand, Proc., Soc. Behav. Sci. 111 (2014) 518–527, https://doi.org/10.1016/j.sbspro.2014.01.085, Transportation: can we do more with less resources? – 16th Meeting of the Euro Working Group on Transportation – Porto 2013.
[17] P. Borgnat, É. Fleury, C. Robardet, A. Scherrer, Spatial analysis of dynamic movements of Vélo'v, Lyon's shared bicycle program, in: Complex Systems Society, ECCS'09, 2009.
[18] P. Borgnat, P. Abry, P. Flandrin, C. Robardet, J.B. Rouquier, E. Fleury, Shared bicycles in a city: a signal processing and data analysis perspective, Adv. Complex Syst. 14 (03) (2011) 415–438, https://doi.org/10.1142/S0219525911002950.
[19] Y. Zhang, M.J. Brussel, T. Thomas, M.F. van Maarseveen, Mining bike-sharing travel behavior data: an investigation into trip chains and transition activities, Comput. Environ. Urban Syst. 69 (2018) 39–50.
[20] K. Gebhart, R. Noland, The impact of weather conditions on bikeshare trips in Washington, DC, Transportation 41 (2014) 1205–1225, https://doi.org/10.1007/s11116-014-9540-7.
[21] W. El-Assi, M.S. Mahmoud, K.N. Habib, Effects of built environment and weather on bike sharing demand: a station level analysis of commercial bike sharing in Toronto, Transportation 44 (3) (2017) 589–613, https://doi.org/10.1007/s11116-015-9669-z.
[22] K. Kim, Investigation on the effects of weather and calendar events on bike-sharing according to the trip patterns of bike rentals of stations, J. Transp. Geogr. 66 (2018) 309–320, https://doi.org/10.1016/j.jtrangeo.2018.01.001.
[23] H. Younes, Z. Zou, J. Wu, G. Baiocchi, Comparing the temporal determinants of dockless scooter-share and station-based bike-share in Washington, DC, Transp. Res., Part A, Policy Pract. 134 (2020) 308–320, https://doi.org/10.1016/j.tra.2020.02.021.
[24] J. Zhang, X. Pan, M. Li, S.Y. Philip, Bicycle-Sharing System Analysis and Trip Prediction, 2016 17th IEEE International Conference on Mobile Data Management (MDM), vol. 1, IEEE, 2016, pp. 174–179.
[25] J. Holmgren, S. Aspegren, J. Dahlströma, Prediction of bicycle counter data using regression, Proc. Comput. Sci. 113 (2017) 502–507, https://doi.org/10.1016/j.procs.2017.08.312.
[26] J. Holmgren, G. Moltubakk, J. O'Neill, Regression-based evaluation of bicycle flow trend estimates, Proc. Comput. Sci. 130 (2018) 518–525, https://doi.org/10.1016/j.procs.2018.04.073.
[27] R. Harikrishnakumar, S. Nannapaneni, Forecasting bike sharing demand using quantum Bayesian network, Expert Syst. Appl. 221 (2023) 119749, https://doi.org/10.1016/j.eswa.2023.119749, https://www.sciencedirect.com/science/article/pii/S0957417423002506.
[28] J.E. Froehlich, J. Neumann, N. Oliver, Sensing and predicting the pulse of the city through shared bicycling, in: Twenty-First International Joint Conference on Artificial Intelligence, 2009.
[29] X. Shi, Y. Wang, F. Lv, W. Liu, D. Seng, F. Lin, Finding communities in bicycle sharing system, J. Vis. 22 (6) (2019) 1177–1192, https://doi.org/10.1007/s12650-019-00587-0.
[30] M. Noussan, G. Carioni, F.D. Sanvito, E. Colombo, Urban mobility demand profiles: time series for cars and bike-sharing use as a resource for transport and energy modeling, Data 4 (3) (2019) 108, https://doi.org/10.3390/data4030108.
[31] Y. Li, Y. Zheng, H. Zhang, L. Chen, Traffic prediction in a bike-sharing system, in: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL '15, Association for Computing Machinery, New York, NY, USA, ISBN 9781450339674, 2015.
[32] L. Lin, Z. He, S. Peeta, Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach, Transp. Res., Part C, Emerg. Technol. 97 (2018) 258–276, https://doi.org/10.1016/j.trc.2018.10.011.
[33] P.C. Chen, H.Y. Hsieh, X.K. Sigalingging, Y.R. Chen, J.S. Leu, Prediction of station level demand in a bike sharing system using recurrent neural networks, in: 2017 IEEE 85th Vehicular Technology Conference, VTC Spring, 2017, pp. 1–5.

[34] A. Faghih-Imani, N. Eluru, A.M. El-Geneidy, M. Rabbat, U. Haq, How land-use and urban form impact bicycle flows: evidence from the bicycle-sharing system (BIXI) in Montreal, J. Transp. Geogr. 41 (2014) 306–314, https://doi.org/10.1016/j.jtrangeo.2014.01.013, https://www.sciencedirect.com/science/article/pii/S0966692314000234.
[35] M. Hu, P. Dai, H. Lin, G. Kong, Improving the station-level demand prediction by using feature engineering in bike sharing systems, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2018, pp. 2103–2108.
[36] A.P. Cortez Ordoñez, P.P. Vázquez Alcocer, Analysis and visual exploration of prediction algorithms for public bicycle sharing systems, in: International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIP2021), Connected Smart Cities (CSC2021), and Big Data Analytics, Data Mining and Computational Intelligence (BIGDACI 2021), Curran Associates, 2021, pp. 61–70, held at the 15th Multi-Conference on Computer Science and Information Systems (MCCSIS 2021) online, 20-23 July 2021.
[37] D. Tomaras, I. Boutsis, V. Kalogeraki, A holistic approach for modeling and predicting bike demand, Inf. Sci. 111 (2023) 102129, https://doi.org/10.1016/j.is.2022.102129, https://www.sciencedirect.com/science/article/pii/S0306437922001077.
[38] Y.C. Yin, C.S. Lee, Y.P. Wong, Demand prediction of bicycle sharing systems, http://cs229.stanford.edu/proj2014/Yu-chun%20Yin%2C%20Chi-Shuen%20Lee%2C%20Yu-Po%20Wong%2C%20Demand%20Prediction%20of%20Bicycle%20Sharing%20Systems.pdf.
[39] Y. Feng, S. Wang, A forecast for bicycle rental demand based on random forests and multiple linear regression, in: G. Zhu, S. Yao, X. Cui, S. Xu (Eds.), 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017, Wuhan, China, May 24-26, 2017, IEEE Computer Society, 2017, pp. 101–105.
[40] B. Wang, I. Kim, Short-term prediction for bike-sharing service using machine learning, Transp. Res. Proc. 34 (2018) 171–178, https://doi.org/10.1016/j.trpro.2018.11.029, https://www.sciencedirect.com/science/article/pii/S2352146518303181, International Symposium of Transport Simulation (ISTS'18) and the International Workshop on Traffic Data Collection and its Standardization (IWTDCS'18), Emerging Transport Technologies for Next Generation Mobility.
[41] S.J. Choi, J. Jiao, H.K. Lee, A. Farahi, Combatting the mismatch: modeling bike-sharing rental and return machine learning classification forecast in Seoul, South Korea, J. Transp. Geogr. 109 (2023) 103587, https://doi.org/10.1016/j.jtrangeo.2023.103587, https://www.sciencedirect.com/science/article/pii/S0966692323000595.
[42] C. Xu, J. Ji, P. Liu, The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets, Transp. Res., Part C, Emerg. Technol. 95 (2018) 47–60, https://doi.org/10.1016/j.trc.2018.07.013, https://www.sciencedirect.com/science/article/pii/S0968090X18306764.
[43] P Hulot, D Aloise, SD. Jena, Towards station-level demand prediction for effective rebalancing in bike-sharing systems, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, ISBN 9781450355520, 2018, pp. 378–386.
[44] Á. Lozano Murciego, J. De Paz, G. Villarrubia, D. Hernández de la Iglesia, J. Bajo, Multi-agent system for demand prediction and trip visualization in bike sharing systems, Appl. Sci. 8 (2018) 67, https://doi.org/10.3390/app8010067.
[45] R.A. Rixey, Station-level forecasting of bikesharing ridership: station network effects in three U.S. systems, Transp. Res. Rec. 2387 (1) (2013) 46–55, https://doi.org/10.3141/2387-06.
[46] A. Cortez-Ordoñez, J.A. Sanchez-Espigares, P.P. Vázquez, A visual tool for the analysis of usage trends of small and medium bicycle sharing systems, Comput. Graph. 109 (2022) 30–41, https://doi.org/10.1016/j.cag.2022.09.009, https://www.sciencedirect.com/science/article/pii/S0097849322001777.
[47] X. Yan, X. Su, Linear Regression Analysis: Theory and Computing, World Scientific Publishing Company, ISBN 9789814470087, 2009, URL https://books.google.se/books?id=afzFCgAAQBAJ.
[48] S.C. Hillmer, G.C. Tiao, An arima-model-based approach to seasonal adjustment, J. Am. Stat. Assoc. 77 (377) (1982) 63–70, https://doi.org/10.1080/01621459.1982.10477767.
[49] R.J. Hyndman, G. Athanasopoulos, Forecasting: Principles and Practice, OTexts, 2018.
[50] T.K. Ho, Random decision forests, in: Proceedings of the Third International Conference on Document Analysis and Recognition - Volume 1, ICDAR '95, IEEE Computer Society, USA, ISBN 0818671289, 1995, p. 278.
[51] S. Taylor, B. Letham, Forecasting at scale, Am. Stat. (2017) 72, https://doi.org/10.1080/00031305.2017.1380080.
[52] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: B. Krishnapuram, M. Shah, A.J. Smola, C. Aggarwal, D. Shen, R. Rastogi (Eds.), KDD, ACM, ISBN 978-1-4503-4232-2, 2016, pp. 785–794, http://dblp.uni-trier.de/db/conf/kdd/kdd2016.html#ChenG16.
[53] P. Madley-Dowd, R. Hughes, K. Tilling, J. Heron, The proportion of missing data should not be used to guide decisions on multiple imputation, J. Clin. Epidemiol. 110 (2019) 63–73, https://doi.org/10.1016/j.jclinepi.2019.02.016, https://www.sciencedirect.com/science/article/pii/S0895435618308710.