2018 IEEE 30th International Conference on Tools with Artificial Intelligence

Bike Usage Forecasting for Optimal Rebalancing Operations in Bike-Sharing Systems

Simon Ruffieux
HumanTech Institute
University of Applied Sciences and Arts of Western Switzerland (HES-SO)
Fribourg, Switzerland
simon.ruffieux@hes-so.ch

Elena Mugellini
HumanTech Institute
University of Applied Sciences and Arts of Western Switzerland (HES-SO)
Fribourg, Switzerland
elena.mugellini@hes-so.ch

Omar Abou Khaled
HumanTech Institute
University of Applied Sciences and Arts of Western Switzerland (HES-SO)
Fribourg, Switzerland
omar.aboukhaled@hes-so.ch

***Abstract*— This article presents the first step of a project focusing on enhancing the management of bike-sharing systems. The objective of the project is to optimize the daily rebalancing operations that need to be performed by operators of bike-sharing systems using machine-learning algorithms and constraint programming. This study presents an evaluation of machine learning algorithms developed for forecasting the availability of bikes on three Swiss bike-sharing networks. The results demonstrate the superiority of the Multi-Layer Perceptron algorithm for forecasting available bikes at station-level for different prediction horizons and its applicability for real-time prediction generation.**

Keywords—machine learning, forecasting, optimal rebalancing, bike-sharing systems, smart mobility, smart city

I. Introduction

The cities of the future will rely on multiple transportation systems. Bike-sharing systems are currently seen as a promising alternative to traditional public transit systems for facilitating short-distance trips inside dense urban locations. Bike sharing reduces traffic, noise, and air pollution and offers users exercise. Therefore, bike-sharing systems have become increasingly popular worldwide with systems deployed in more than 2000 cities around the world [1]. The technology of bike-sharing systems constantly evolves in order to facilitate management and user satisfaction; from docked to free-floating systems, from standard bikes to electric bikes, etc. The management and operating of those systems is not trivial as it implies ensuring the number of bikes in the system and their distribution amongst the stations always satisfies the user demand. Most of the cost of operating current bike-sharing systems originates from the personnel required to collect bikes from overcrowded stations and redistribute them equally amongst the stations suffering a shortage of bikes [2]. Those rebalancing operations are usually performed several times daily by specific teams of field operators using small trucks or carrier-bikes.

In a collaboration with a Swiss company managing bike-sharing systems (Intermobility SA), we are developing a framework to optimize the operations required to maintain the bike network in a balanced state. The proposed system relies on forecasts of bike usage. A system has been developed to forecast the number of bikes available at the different stations of a network for different future horizons and to optimize the routing required to rebalance the network relying on those predictions. This paper describes the whole system and presents the initial results of the forecasting system using the real-world data collected from three cities located in Switzerland.

This paper is structured as follows. Section II provides a review of the related works on shared-bike systems with a focus on forecasting. Section III present an overview of the architecture of the framework to clarify the context of the study. Section IV details the prediction system and the methodology used to quantify the performances of the evaluated algorithms. The Section V presents the results of the evaluations and the Section VI concludes the study.

II. State of the Art

With the increase of bike-sharing systems worldwide and many bike-sharing companies providing an open access to historical and real-time data from their networks; numerous studies on bike-sharing systems have been published these last years. These studies have covered several topics related to bike-sharing systems briefly summarized hereafter. The availability of open data also fostered new studies on optimizing rebalancing and routing operations and on algorithms for forecasting demands and/or bikes availability.

In their work, Chen et al. have proposed a recommender system to infer the optimal placement of bike docking stations based on prediction of localized demand using heterogeneous urban open data [3]. Their system outperformed state of the art approaches and reduced the need for urban planners. In their study, Jimenez et al. investigated metrics to characterize the effectiveness of existing stations in a network [4]. They proposed the "Turnover station ratio" metric that measures the number of times a station's capacity is used in a complete day. Studies also investigated how to quantify and improve user satisfaction. Kaspi et al. found out that faulty bikes and empty/full stations have the most impact on user dissatisfaction [5]. Vassimon performed a benchmark analysis of the key factors impacting bike-sharing system in more than 50 cities [6]. Higher-level studies have also been performed such as Razzaque and Clarke who envisioned the next generation of bike-sharing systems through a new communication infrastructure [7]. Their proposed system would collect and process real-time data about

2375-0197/18/$31.00 ©2018 IEEE
DOI 10.1109/ICTAI.2018.00133

customers, environment and bikes to provide new services to customers and companies. Semtech's LoRa low-power wireless communication technology is currently being largely adopted by bike-sharing companies [8] and should probably lead the market to the Internet-of-Bikes.

Several studies investigated rebalancing bike networks and the optimal routing operations required to perform this rebalancing. Two type of methods are distinguished: static and dynamic rebalancing. The static approach considers the network being static once the algorithms starts its computations until actual operations on the field are finished [9]–[11]. These approaches usually consider a single repositioning operation at the beginning of day. Dynamic approaches consider the evolution of the network over time and can constantly update the rebalancing tasks and routes in real-time. Such solution assumes rebalancing operators are constantly available [12]–[14]. Most approaches divide the problem into two inter-dependant steps, finding the redistribution tasks (which bikes must be collected and in which stations to drop them) in order to achieve the best balance and finding the optimal routes to perform the tasks according to the number of vehicles available. Most of the studies reviewed focus on optimizing the operational cost for the bike company by reducing the unmet demand (stations with no bikes) [15].

Forecasting of demand, bike availability or empty stations has been investigated in several studies these last years. Due to the type of open datasets available, several studies focused on forecasting at the global network level. In their study, Giot and Cherrier notably highlighted the good performances of Ridge Regression to forecast hourly bike usage of a network up to 24 hours ahead [16]. Dias et al. focused on predicting full or empty stations up to two days ahead using ARIMA and Random Forest algorithms [17]. In their study, Random Forest outperformed ARIMA. Yoon et al. developed a journey advisor using 5 and 60 minutes forecasts of bikes availability at station-level using an ARIMA algorithm based on seasonal trends and spatial correlations [18]. Yang et al compared several algorithms to predict pick-up 30 minutes ahead and proposed a novel mobility model that characterizes spatiotemporal transitions of bikes amongst stations to estimate drop-off at station-level. Their study demonstrated the superior performances of the Random Forest algorithm for pick-up prediction compared to ARMA and Historical Average naive approaches [19]. In a previous study, we investigated the impact of the modelling of the data on accuracy performances of a Random Forest algorithm and compared the best solution with an approach based on Convolutional Neural Network [20]. The study demonstrated the importance of modeling each station of a network individually and the beneficial impact of taking into account spatiotemporal data for predictions. Furthermore, the results highlighted the performances of Random Forest on predictions up to 6 hours ahead.

In a recent study, Liu et al. proposed a solution similar to the framework presented in this study [21]. They proposed a custom K-Nearest-Neighbour algorithm to predict hourly pick-up and drop-off demands based on meteorological conditions and then use this information to define the target inventory level of each station in order to feed their rebalancing algorithm. However, according to our partner company, most of the works on rebalancing presented above would not be directly applicable in their real-world context, as many requirements are not satisfied; such as limited number of rebalancing operators, limited schedule of operators, reduced capacity of carrier-vehicles, distributing the work equally amongst the operators, etc.

III. System Overview

In the project, a whole framework is developed to facilitate the management of field operations for bike-sharing companies. The proposed framework is represented in Fig.1. It consists in a 3-layer architecture: data collection, algorithms management and visualization interface.

In the *Data-layer*, a program collects the data from the monitored bike network and weather data every 5 minutes from web-services and adds them on a local PostgreSQL DB, thus generating a *Historical database*. The bike data is directly collected from the different bike networks managed by the Intermobility company (IMY). The weather data is collected from Open Weather Map (OWM); which is a free web-service providing current and forecast meteorological conditions for many cities around the world. In the *Algorithm-layer*, the prediction system automatically generates new predictions for each of the monitored networks every 5 minutes and stores them in the Prediction DB. The rebalancing system generates a set of rebalancing tasks for the field operators upon request of the network manager. The rebalancing system relies on current and forecasted information of the network. Using the *Visualization interface*, in the form of a dedicated website, the bike network manager can visualize the current and forecasted status of each station of the network and create custom rebalancing requests with specific parameters: number of available operators, area of interests, etc. The field operators can visualize the rebalancing missions that needs to be achieved with, for each mission, the list of subtasks that must performed according to the optimal routes provided by the rebalancing algorithm. Each field operator can monitor his progress for his current mission.

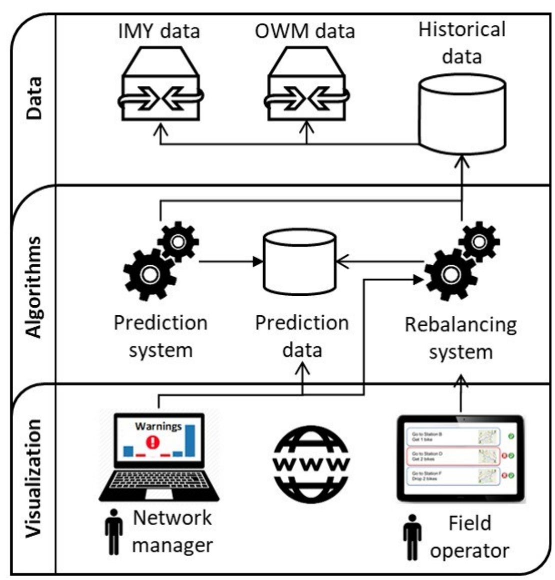

Fig. 1. Schematic overview of the 3-layer architecture of the proposed framework.

This article focuses on the evaluation of the prediction system at forecasting accurately the number of bikes present in the stations of a network at different temporal horizons.

IV. Prediction System

The prediction system is a key-component of the complete framework. Its accuracy is important for bike-sharing companies to be able to detect in advance potential empty stations and to feed accurate forecasts to the rebalancing system. Following discussions with a partner bike-sharing company, the system should be usable shortly after a new bike network has been deployed in a new city, which means that the system must be able to run with limited quantity of training data. The system should also provide the possibility to update its knowledge periodically (monthly) with the latest available data in order to adapt to topological and user behavioural changes in the bike network. One constraint is also the possibility to execute the developed algorithms to generate real-time predictions on affordable hardware owned by company.

A. Conception

The *prediction system* has been developed following most of the concepts defined in a previous study [20]. A schematic overview of the system is provided in Fig. 2. The system uses one model of algorithm per station. The input features for each model consist of data from the station itself (#bikes, #departures and #arrivals at current and 4 previous time steps), from related neighbour stations (#departures at 4 previous time steps), from temporal information (hour, day, month and holiday type) and from environmental data (temperature, humidity, wind at current time and for a 3 hour forecast. Holiday type corresponds to 'no holiday', 'school' or 'public' holiday. The related neighbour stations correspond to the 5 stations pouring most bikes into the current station, as inferred from training data (the concept is similar to ISBT stations from [21]).

As illustrated in the Fig.2, the prediction system has been developed specifically for the context of bike-sharing systems. Its modular structure provide the possibility to change easily the type of algorithms or the pre-processing operations to evaluate different solutions. The system loads the raw data from the historical database through a SQL request and performs pre-processing operations to correct missing time steps and/or information and to format the shape of the data for the selected machine-learning algorithm.

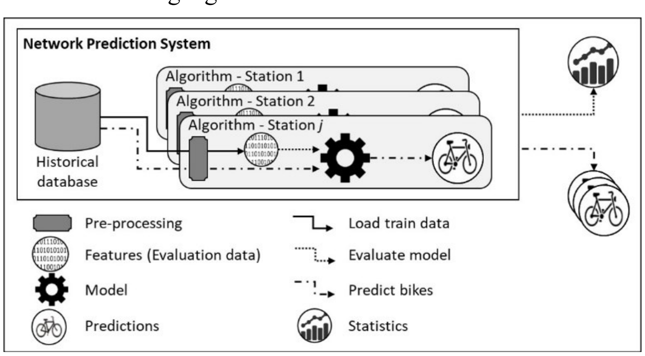

Fig. 2. Schematic view of developed prediction system with one model of algorithm per station.

The system can be used in evaluation mode or in real-time prediction mode. In the evaluation mode, the system collects historical data for a defined period and splits it into 80% for training (oldest data) and 20% for testing (most recent data); the algorithm is then trained on 80% and evaluated on the remaining 20%. Once all the stations of a network have been trained and evaluated, a detailed report with various statistics is generated. In real-time mode, only the most recent measures are collected from the database and predictions are generated using the previously trained/saved model. The system can be parametrized to generate predictions for multiple horizons. The experiments presented in this article use the following horizons (in minutes): [15, 30, 60, 120, 360, 720]. According to the reviewed literature and constraints of the project, two different algorithms have been selected for this study: Random Forest (RF) and Multi-Layer Perceptron (MLP). Both algorithms have been implemented in Python using the scikit learn machine-learning framework using the provided *RandomForestRegressor* and *MLPRegressor* classes [22].

B. Evaluation methodology

A precise methodology has been defined for the evaluation of the selected algorithms. The two algorithms selected use the exact same set of input features and are evaluated using the same training and test data. For each algorithm, an initial evaluation is performed with default scikit–learn parameters; then a GridSearch is performed in order to find the best hyper-parameters. Finally, a final evaluation is conducted with the best set of hyper-parameters. In this study, the results given by the best set of hyper-parameters are reported in terms of root-mean-squared error (RMSE).

The algorithms are compared using different metrics. The most important metric, RMSE, measures the accuracy of the predictions. Execution time and size in memory of the different algorithms are also investigated to ensure that the proposed algorithm respects the real-time prediction constraint of the project. The RMSE has been chosen for the accuracy metric as it penalizes more heavily large errors, which is highly desirable in the context of the bike-sharing. The RMSE is defined in (1), where n is the number of stations in a network and y_j and $\hat{y}_j$ are respectively the real and predicted number of bikes in the station j at the forecasted horizon t:

$$RMSE(t) = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_{j(t)} - \hat{y}_{j(t)}\right)^2} \qquad (1)$$

The evaluation duration corresponds to the raw data loading duration plus the training and testing durations for a whole network. The prediction duration corresponds to the raw prediction data loading duration and prediction generation for a whole network. The models size corresponds to the size in memory of all models of the network. The prediction duration is a critical metric in order to assess real-time capabilities of the system.

The algorithms are compared with the results of two naïve solutions: the *Persistence* and *Mean Hour* models. These models are frequently used in literature on forecasting as a basis for comparisons. The *Persistence* model uses the last known number of bikes as the predicted value for all future horizons.

The *Mean Hour* model uses the average number of bikes at a specific hour as the predicted value. The averages are inferred directly from the training data.

The evaluation has been performed on three different networks managed by Intermobility SA; each corresponding to a Swiss city. Each considered network has different characteristics influencing the evaluation results. Bienne is a medium-size city and its bike-sharing network is composed of 52 stations regularly used by the citizens. Vevey is a small-size touristic city located along Lake Geneva and its network is composed of 17 stations set along the coast of the lake, forming an elongated shape. Geneva is a large-size city and its network is composed of 80 stations dispatched in several suburban locations, forming several distinct clusters. Geneva network is little used. The usage of the stations from those three networks is disparate; some stations being used very often and others rarely. The impact of this disparity will be briefly discussed in the presentations of the results.

All evaluations were performed using the historical data collected every 5 minutes from the 28.02.2018 to the 30.05.2018. All experiments were executed on a computer running Windows 10, Python 3.5.4 and scikit-learn 0.19.1 on an Intel i7-3820 3.6 GHz with 4 cores and 16 GB of RAM.

V. RESULTS

As mentioned in the previous section, the RMSE is reported for the two naïve models and two algorithms: Persistence, Mean Hour (MH), Random Forest (RF) and Multi-Layer Perceptron (MLP). Except for the MH model, it is expected that the RMSE will decrease along with the increase of the predicted horizon due to the introduction of higher uncertainty.

GridSearch was performed for both algorithms. For the Multi-Layer Perceptron, 162 combinations of hyper-parameters were evaluated in approximately 100 hours. The best set of parameters yielded an improvement of 5.7% compared to the default parameters (RMSE 0.9031->0.8518). For the Random Forest, 42 combinations of hyper-parameters were evaluated in approximately 5 hours and the best set of parameters yielded an improvement of 14.3% (RMSE 1.0884->0.9325). The best set of parameters found for each algorithm are described below. Refer to the scikit-learn documentation for a detailed description of each parameter.

- **Random Forest**: n_estimators = 500, max_features ='sqrt' and min_samples_leaf = 50
- **Multi Layer Perceptron**: learning_rate_init = 0.01, max_iter = 1000, activation='relu', learning_rate = 'adaptive', solver = 'adam', alpha = 0.0001, hidden_layer_sizes = (16,), early_stopping = True

The detailed results of the evaluation of the best model on the bike-sharing network of Bienne are presented in Fig.3. The MLP outperformed the naïve models on every prediction horizon, although the gain is smaller for short prediction terms. The RF was outperformed by the Persistence model in average and specifically on short-term horizons. The MLP largely outperformed the RF algorithm. The difference of performances between the two algorithms reduces with increasing prediction

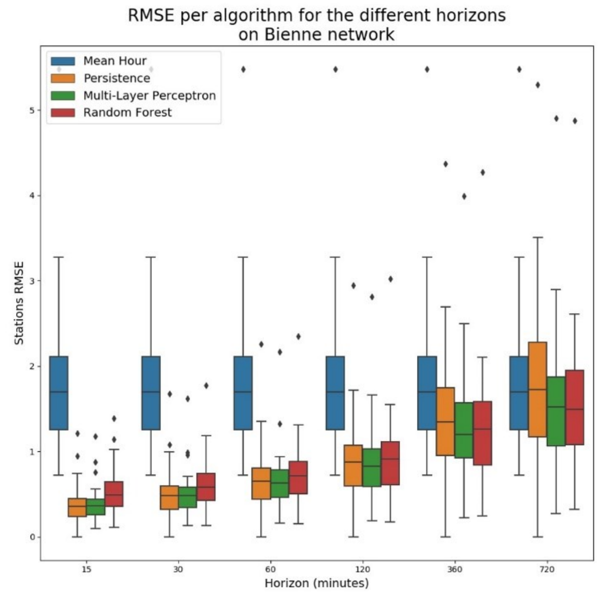

Fig. 3. Detailed comparative results of the different algorithms evaluated on the Bienne network.

terms and RF slightly outperforms MLP for the 720 minutes horizon. In Fig.3, the presence of outliers (black diamonds) can be observed; they correspond to station(s) with high turnover rate. Their usage pattern is generally harder to predict and subject to more variations, hence a larger prediction error. The largest outlier corresponds to the station situated next to the railway station in Bienne. Separate GridSearch evaluations were performed focusing only on those outliers in order to infer if improvement was possible but similar performances were obtained.

The summative results in terms of processing duration and size of models are presented in TABLE I. A large portion of the reported durations corresponds to the loading of the raw data from the historical database (620 seconds for the evaluation). This operation requires multiple MERGE and JOIN operations for each measure, yielding long delays. For the prediction operation, these requests have been parallelized in order to minimize the total duration. The requests still last about 32 seconds and should probably be further optimized. The prediction processing time is ~11 seconds for MLP and ~19 seconds for RF. The size in memory of the RF models is considerably larger than MLP, notably due to the difference in terms of model complexity (*n_estimators* vs *hidden_layer_sizes*). These results validate the feasibility of generating pseudo real-time predictions every 5 minutes.

TABLE I. SUMMARY OF PROCESSING TIMES ON THE BIENNE NETWORK

Results	**Processing times (seconds) and model size (MB)**		
	Evaluation duration	***Prediction duration***	***Model size***
Random Forest	730 (620,110)	50.88 (32, 19)	31'200
Multi-Layer Perceptron	702 (620, 82)	42.81 (32, 11)	1.5

Finally, the summative results of the algorithms for the three networks are presented in TABLE II. The results demonstrate the superiority of the MLP algorithm. The decreasing RMSE of the Persistence model on each of the network (Bienne > Vevey > Geneva) can be correlated with the network usage by customers. The RMSE improvement obtained by the MLP compared to the Persistence model tends to demonstrate that networks with higher usage gain more benefit from MLP. Surprisingly, the RF algorithm was slightly outperformed by the Persistence model. These results should be further investigated with more training data as RF achieved the best results in many of the reviewed studies.

TABLE II. SUMMARY OF EVALUATION RESULTS USING THE BEST SET OF PARAMETERS FOR EACH NETWORK

Results	Average RMSE		
	Bienne	***Vevey***	***Geneva***
Persistence	0.9159	0.7991	0.3038
Mean Hour	1.7788	1.8569	0.9262
Random Forest	0.9325	0.8403	0.3425
Multi-Layer Perceptron	**0.8518**	**0.7725**	**0.2949**

It is important to note that the project has the constraint to operate with a limited period of training data; this might have influenced the performances obtained in this study. The small hidden layer topology of the best MLP model might notably be linked to this limited quantity of training data used for the evaluations.

VI. CONCLUSION

In this study, a complete framework to facilitate the management of field operations for bike-sharing systems has been presented. The *Prediction System* developed to forecast the quantity of bikes available at station-level for different horizons has been detailed. The study provides a comparative analysis of the results of Multi-Layer Perceptron, Random Forest, Persistence and Mean Hour algorithms in terms or forecasting accuracy for different prediction horizons. The results were validated with the data collected on three different bike-sharing network in Switzerland. The Multi-Layer Perceptron algorithm obtained the best results in terms of forecasting accuracy and processing performances.

In the future, additional algorithms such as Long Short Term Memory, Gate Recurrent Unit and Extreme Gradient Boosting should also be evaluated and notably with various amounts of training data in order to better characterize their performances on long-term. A custom rebalancing system will be developed to fulfil the specific field requirements of Intermobility SA. The two system will then be connected together and linked to the visualization interface allowing the management of the system for bike companies and field operators. Finally, the efficiency of the whole framework will be evaluated in real conditions in collaboration with the company on the three considered bike-sharing networks.

ACKNOWLEDGMENT

The authors would like to thank Intermobility SA for sharing their data and their knowledge about field operations in bike-sharing systems. The project is funded by the Swiss Agency for Innovation Promotion (Innosuisse), KTI P-Nr.: 26149.2 PFES-ES.

REFERENCES

[1] "The Bike-sharing Blog." [Online]. Available: http://bike-sharing.blogspot.com/. [Accessed: 30-May-2018].

[2] J. Büttner, H. Mlasowsky, and T. Birkholz, *Optimising Bike Sharing in European Cities*, unpublished, 2011.

[3] L. Chen *et al.*, "Bike sharing station placement leveraging heterogeneous urban open data," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*, 2015, pp. 571–575.

[4] P. Jiménez, M. Nogal, B. Caulfield, and F. Pilla, "Perceptually important points of mobility patterns to characterise bike sharing systems: The Dublin case," *J. Transp. Geogr.*, vol. 54, no. October, pp. 228–239, 2016.

[5] M. Kaspi, T. Raviv, and M. Tzur, "Bike-sharing systems: User dissatisfaction in the presence of unusable bicycles," *IISE Trans.*, vol. 49, no. 2, pp. 144–158, Feb. 2017.

[6] P. P. De Vassimon, "Performance Evaluation for Bike-Sharing Systems : a Benchmarking among 50 Cities,", unpublished, 2016.

[7] M. A. Razzaque and S. Clarke, "Smart management of next generation bike sharing systems using Internet of Things," in *2015 IEEE First International Smart Cities Conference (ISC2)*, 2015, pp. 1–8.

[8] N. Blenn and F. Kuipers, "LoRaWAN in the Wild: Measurements from The Things Network," *ArXiv*, Jun. 2017.

[9] J. Schuijbroek, R. C. Hampshire, and W.-J. van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *Eur. J. Oper. Res.*, vol. 257, no. 3, pp. 992–1004, Mar. 2017.

[10] W. Y. Szeto and C. S. Shui, "Exact loading and unloading strategies for the static multi-vehicle bike repositioning problem," *Transp. Res. Part B Methodol.*, vol. 109, 2018.

[11] D. Chemla, F. Meunier, and R. Wolfler Calvo, "Bike sharing systems: Solving the static rebalancing problem," *Discret. Optim.*, vol. 10, no. 2, pp. 120–146, 2013.

[12] F. Chiariotti, C. Pielli, A. Zanella, and M. Zorzi, "A dynamic approach to rebalancing bike-sharing systems," *Sensors (Switzerland)*, vol. 18, no. 2, pp. 1–22, 2018.

[13] S. Ghosh and P. Varakantham, "Incentivizing the Use of Bike Trailers for Dynamic Repositioning in Bike Sharing Systems," no. International Conference on Automated Planning and Scheduling, pp. 373–381, 2017.

[14] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic repositioning to reduce lost demand in bike sharing systems," *J. Artif. Intell. Res.*, vol. 58, pp. 387–430, 2017.

[15] S. Ghosh, M. Trick, and P. Varakantham, "Robust repositioning to counter unpredictable demand in bike sharing systems," in *IJCAI International Joint Conference on Artificial Intelligence*, 2016.

[16] R. Giot and R. Cherrier, "Predicting bikeshare system usage up to one day ahead," in *2014 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*, 2014, pp. 22–29.

[17] G. M. Dias, B. Bellalta, and S. Oechsner, "Predicting occupancy trends in Barcelona's bicycle service stations using open data," *IntelliSys 2015 - Proc. 2015 SAI Intell. Syst. Conf.*, pp. 439–445, 2015.

[18] J. W. Yoon, F. Pinelli, and F. Calabrese, "Cityride: A Predictive Bike Sharing Journey Advisor," *2012 IEEE 13th Int. Conf. Mob. Data Manag.*, pp. 306–311, 2012.

[19] Z. Yang, J. Hu, Y. Shu, P. Cheng, J. Chen, and T. Moscibroda, "Mobility Modeling and Prediction in Bike-Sharing Systems," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '16*, 2016, pp. 165–178.

[20] S. Ruffieux, Spycher Nicolas, E. Mugellini, and O. Abou Khaled, "Real-Time Usage Forecasting for Bike-Sharing Systems A Study on Random Forest and Convolutional Neural Network Applicability," in *SAI Intelligent Systems Conference*, 2017.

[21] J. Liu, L. Sun, W. Chen, and H. Xiong, "Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016, pp. 1005–1014.

[22] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.