Free-Floating Micro-mobility Smart Redistribution Using Spatio-temporal Demand Forecasting

Rania Swessi, Zeineb El Khalfi, Imen Jemili, Mohamed Mosbah

▶ **To cite this version:**

Rania Swessi, Zeineb El Khalfi, Imen Jemili, Mohamed Mosbah. Free-Floating Micro-mobility Smart Redistribution Using Spatio-temporal Demand Forecasting. 2023 IEEE Vehicular Networking Conference (VNC), Apr 2023, Istanbul, Turkey. pp.73-80, 10.1109/VNC57357.2023.10136284 . hal-04114850

HAL Id: hal-04114850
https://hal.science/hal-04114850v1

Submitted on 2 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Free-Floating Micro-mobility Smart Redistribution Using Spatio-temporal Demand Forecasting

Rania Swessi
Faculty of sciences of Bizerte
University of Carthage, Tunisia
rania.swessi@fsb.ucar.tn
ORCID 0000-0003-1107-4673

Zeineb El Khalfi
CESI Campus de Bordeaux
LINEACT CESI, France
zelkhalfi@cesi.fr
ORCID 0000-0002-9496-768X

Imen Jemili
RIADI, ENSI
University of Manouba, Tunisia
imen.jemili@fsb.ucar.tn
ORCID 0000-0002-3701-1251

Mohamed Mosbah
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, France
mohamed.mosbah@u-bordeaux.fr
ORCID 0000-0001-6031-4237

***Abstract*—Micro-mobility refers to a variety of small and lightweight vehicles designed for individual use. These new vehicles are inexpensive, simple to operate, and enjoyable to ride, making them the easiest and most suitable mode of transportation for trips of less than five miles. They have become extremely popular, especially with the advent of free-floating systems that offer users flexible parking in order to facilitate the rental process. However, the problem of imbalance and maldistribution is among the major challenges of these systems, causing dissatisfaction and loss of customers. Therefore, to ensure the balancing of the fleet and to make the best decision for its reorganization, we must consider strategic locations that are accessible to all. In this paper, we propose a machine learning model for spatio-temporal demand forecasting using a multi-output regression technique. The main goal of the paper is to help pick the ideal areas for fleet deployment and balance the system according to user needs. Our solution, designed for public electric scooters, is based on the estimation of user demand over a grid-based service area. In addition, we propose an enhanced solution that outperforms other baseline models, including the Random Forest, Gradient Boosting, and Stacking Regressor.**

***Index Terms*—micro-mobility, free-floating, maldistribution problem, repositioning, spatio-temporal demand forecasting, machine learning, multi-output regression, time-series forecasting**

I. Introduction

Nowadays, micro-mobility services have gained great popularity. E-scooters, bikes, or motorcycles are examples of micro-mobility devices that offer an attractive solution to minimize car dependency while helping cities achieve their social and environmental goals and improving road safety [1]. In fact, these lightweight travel vehicles are receiving significant attention due to their ecological character and their ability to solve problems such as traffic jams and increased carbon emissions [1]. Many companies are taking full advantage of the shift in people's perceptions towards this new means of transportation; bikes or e-scooters are used as a shared resource between several users, hence the term "shared micro-mobility". Thanks to their market dominance, these companies provide rental services and improve operational efficiency. Except for a few minor structural variations, each of these companies has its own rental strategy and fleet structure. For the parking procedure, we generally have two operating modes: (i) a dock-based system, where the user is required to park at the nearest station in the service area, which facilitates the management of micro-vehicles; (ii) a free-floating system, which allows users to park the fleet freely, making the rental procedure more convenient and easier [2]. Despite being the most convenient mode, free-floating poses many regulatory challenges such as poorly parked fleets by users, maintenance difficulties, and unloaded fleet collection [3]–[5]. Furthermore, because of the uncontrolled parking system, fleets can be parked in irrelevant areas, leading to an unbalanced system.

Fleet maldistribution or imbalance is one of the major problems that has a negative impact on usage rates and the quality of the user experience within these types of systems. It refers to the unavailability of services at a given time in some regions or specific areas, i.e., fleets are not placed in proper zones where there is a real need for services [6], [7]. Therefore, users are experiencing unsatisfactory services when there are no nearby fleets available and suffer from the distance they have to travel to find a micro-vehicle [7]. As a result, micro-mobility companies are losing an important customer base, which can have a significant impact on the company's revenue. Thus, improving user satisfaction is one of the goals for micro-mobility operators.

To remedy this free-floating maldistribution problem, micro-mobility operators must make fleet repositioning operations to balance the system, which are often handled by trucks or trailers driving around the city and moving fleets among areas. These redistribution operations are usually performed in a manual manner during the system idle periods (such as midnight or early morning) based on points of interest such as subway stations and universities or based on predefined criteria like population density [3], [8]. However, as the demand for these services continues to grow steadily, these static redistribution plans are no longer effective, allowing

This work has been supported by the industrial research Chair "Mobility and Intelligent Transports" (Foundation of Bordeaux University, 2019 - 2023).

for unevenness and imbalance in distributed fleets. Moreover, it is not sufficient to redistribute statically according to the type of area; in fact, it has been shown that variable criteria such as day of the week, temperature, or wind speed are important factors that have a significant impact on the use of micro-mobility services [6], [9], [10]. Thus, automatic redistribution methods that are based on pick-up demand prediction within several zones can be an effective solution for repositioning fleets based on user needs. This method can assist the operators in enhancing the quality of their decision-making since it is increasingly difficult for them to make an appropriate fleet relocation decision manually by examining all the variable criteria. Therefore, conducting a comprehensive spatio-temporal short-term prediction taking into account weather criteria over several spatial areas is of great importance to the operator, who can move the fleets from the areas with less demand to those with higher potential user demands. These automated procedures not only assist to balance supply and demand throughout different time periods, but they also help maximize fleet utilization and decrease the search time to find a fleet. In this context, studies have been conducted to improve these redistribution strategies and increase the service availability based on fleet mobility models and forecasting using machine learning techniques [5], [7], [11]–[15]. However, some studies do not provide an estimate of the exact number of future demand within regions where a fleet deployment is needed [13]–[15] or are computationally complex [7], [12]; in addition, they take into account only a few variable criteria which affect the model performance [5], [7], [12], [14].

In this regard, we propose an effective prediction tool for free-floating systems, based on pick-up demand forecasting from real recorded departure trips, taking into account relevant variable criteria in order to improve fleet distribution and repositioning. Our solution provides micro-mobility operators with future demand predictions, helping them select the best fleet location; in this way, it will be easier for users to find a fleet that improves their journey experience. To deal with the problem of the uncontrolled parking procedure, our strategy consists of using a grid over the operational service area to specify the zones with corresponding predictions in which the operator could deploy fleets. The key contributions of this paper are listed below:

- A grid-based spatial framework was established as part of the preprocessing phase.
- An experimental study based on demand forecasting within zones using classical machine learning algorithms is conducted; a visualization of results is also provided.
- A time-series forecasting solution is proposed to improve our prediction results.

The rest of this paper is organized as follows: Section II gives an overview of relevant literature focusing on free-floating shared micro-mobility redistribution. In Section III, we formulate the proposed approach and its different steps. Section IV presents a case study based on e-scooters' trip data in the city of Chicago. Section V concludes the paper and mentions future works.

II. Literature review on free-floating shared micro-mobility redistribution

The availability of real data from publicly accessible shared micro-mobility systems has helped researchers to examine operational problems using demand forecasting, with a special focus on the imbalance and maldistribution issue [3], [6], [7], [11]–[15]. Notwithstanding, demand prediction in free-floating systems is very challenging due to the characteristics of spatial distributions and the uncontrolled parking procedure since the system counts hundreds of parking positions every day. Dividing the whole service area into small-scale spatial units can be an effective solution to make the prediction.

In [5], the use of a square grid on the service zone is proposed as a solution to identify the spatial units by applying a Long Short-Term Memory (LSTM) for the demand forecasting aspect. It consists of a service zone partition into grid cells, using a field of 9 km^2 where the model can predict the future N steps for each cell of the grid. Based on the obtained results, the system invites customers to group low-load bikes to facilitate the pick-up process and balance the system. This model is trained using a trip database recorded in China in 2017. However, the time training period is limited (less than one month), which does not ensure the model's performance. In [6], a Multi-Layer Perceptron model has been implemented to predict hourly demand within a spatial region partitioned into 20 x 20 grids. Several algorithms, including Convolutional LSTM (Conv-LSTM) and XGBoost, were compared on the same data set collected in Beijing, China in 2017. However, the model's performance in terms of error is low. Using almost the same strategy of gridding, an LSTM model has been implemented in [11] to predict the level of hourly demand within grid cells. The spatial unit was set to a 200 m square grid for the prediction. The solution allows two types of prediction: (i) a prediction within each grid cell; (ii) a prediction within 5 clusters that have been made from grid cells (commercial areas, tourist areas, etc.), for an operator-based rebalancing. Although the solution provides good prediction results with accuracy, the model's performance, in terms of computing time for prediction, is relatively low. The solution took 53$\sim$78 s for the five clusters, and 5700$\sim$6390 s for 1164 square grids, indicating that it cannot be deployed in real-time. Moreover, it does not allow a comparison with other machine learning models on the used dataset in comparison with other studies.

In [14], the authors exploited the heatmap image prediction strategy to identify the most future frequent zones for the fleet. They proposed a Fully Convolutional Network (FCN) to forecast user demand by processing historical demand density images. The FCN model architecture is modified by adding the masking process to better guide the model. The input to the Masked Fully Convolutional Network (MFCN) is a series of historical images representing the demand density at each hour. The proposed model is trained on a dataset of electric scooter

usage over a 75-day period (2019) for the city of Calgary, Canada. In [15], a combination of LSTM and Conv-LSTM is proposed as a solution to predict short-term demand using heatmap images for the demand characteristics in accordance with weather and temporal criteria. The LSTM model is used to capture temporal and weather features, while spatial dependencies are discovered from heatmap images using the Conv-LSTM model. To identify the zones, the solution relies on a grid on the service zone. To ensure the performance of the suggested solution, the study allows a comparison of different algorithms on the same dataset, such as convolutional neural networks and artificial neural networks. Figure 1 represents an example of the model's output.

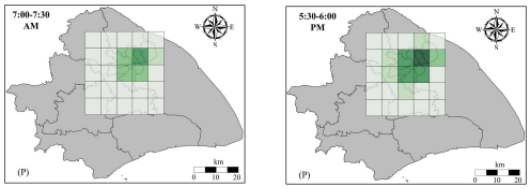

Fig. 1. Example of predicted demand [15]

Using a similar approach, a recent work proposes a solution in [13] that aims to predict the hourly user demand with high spatial resolution using an Encoder-RNN-Decoder (ERD). To efficiently extract latent features from an input heatmap image, the solution relies on autoencoders instead of using a masking process with a weighted loss. The algorithm's structure is mainly composed of 3 large parts: an encoder for features extraction, the Recurrent Neural Network (RNN) cells for predicting future arrivals, and a data decoder. The study is based on real-world collected data from the city of Chungju, South Korea in 2019. Nonetheless, the use of heatmap images for prediction in free-floating systems as a method does not allow for a precise understanding of the user demand density or to have an idea about the exact number of fleets that should be placed in specific areas [13]–[15]. In addition, some algorithms based on RNNs, like ERD, require a significant number of trainable parameters, which increases the complexity of the algorithm [13]. In [7], an automatic balancing system based on deep reinforcement learning[1] framework is proposed using a new algorithm namely Hierarchical Reinforcement Pricing (HRP). The solution consists of formulating the problem as a Markov decision process with the use of the Gated Recurrent Unit (GRU) algorithm for estimating future action values. However, the solution lacks real-environment testing. In [12], a multi-agent system based on deep reinforcement learning, capable of proposing convenient alternative locations for each reservation request, is proposed. In order to maximize the availability of vehicles and minimize the number of relocation operations and battery replacements as well, the system proposes two types of rebalancing; operator-based rebalancing and customer-based rebalancing. Since the solution is based on a simulated environment, the number of iterations and actual data are very few to evaluate the model's performance. In addition, the use of reinforcement learning in the context of an automatic redistribution system is computationally complex.

With our proposed approach, we aim to implement a solution with a clearer prediction output than heatmap images and a lower computing cost based on a real-world database. Besides, to meet the client's needs, we offer a prediction per period for a semi-dynamic balance and a prediction per hour using an advanced method.

III. Study of Free-Floating Smart Redistribution

Our main goal is to study free-floating fleet redistribution, based on forecasting future pick-up user demand using well-defined criteria and historical trip data. We aim to assist micro-mobility operators in making the best fleet redistribution and picking the ideal areas for fleet deployment. Indeed, optimal fleet locations contribute to improved service quality and boost the number of satisfied users by minimizing user travel distances to fleets. To this end, we propose a model based on multi-prediction to estimate the number of future pick-up demands based on user trips within each zone. To have accurate estimations, we partition the service area into hexagonal grid cells; each cell represents a specific zone. In this context, we propose a Machine Learning-based tool for fleet ReDistribution and Optimization (ML-RDO). Then, we introduce an enhancement of this approach. In the following, we detail the two approaches.

A. Machine Learning-based tool for fleet ReDistribution and Optimization (ML-RDO)

ML-RDO relies on the prediction of user demands to allow operators to balance their fleets and improve availability for customers. As exposed in Figure 2, this method consists of 4 steps: Data cleaning and preparation, dataset splitting, models construction and training, models evaluation, and exploitation.

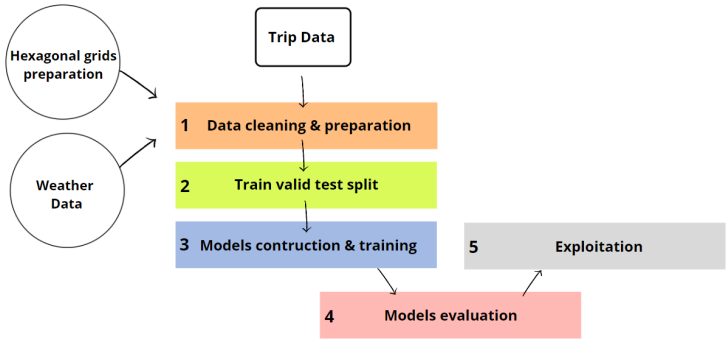

Fig. 2. ML-RDO Overview

1) Data cleaning and preparation: Our data cleaning steps entail removing duplicate and missing values from weather and trip datasets, it also ensures data join and data encoding [16]. Then, we proceed with the preparation phase of the

[1]Deep reinforcement learning involves the training of artificial agents to interact with their environments and learn the best actions to take using a deep neural network to increase reward and solve the problem.

prediction environment. Our strategy is based on hexagonal gridding of the operational service area. In fact, previous studies generally divide the service zone into many sub-zones, mainly using the square shape to predict scooter demand for free-floating systems [5], [11], [13]–[15]. However, this latter shape has several disadvantages, including an imprecise definition of the nearest neighborhood [17]. In another context, researchers have shown that the hexagon shape, which has six sides, is particularly interesting since it covers a plane with units of equal size and leaves no wasted space. Likewise, when compared to square cell shapes, all the neighbors of a hexagon are equally spaced apart, and have softer and smoother gradients that make data interpretation much easier [17]–[19]. Taking into consideration these advantages, our operational zone is divided using the hexagonal shape with the aid of Uber's Hexagonal Hierarchical Spatial Index (H3) library in python.

2) Train-valid-test split: The database is divided into 80% learning data (consisting of a set of labeled instances that serve as learning examples), 10% validation data (used for setting hyperparameters), and 10% to test the model's performance.

3) Models construction: In order to avoid creating several models for each cell of the grid, which is impractical, we fully employ the multiple output regression approach to forecast user demand for micro-vehicles within all the created zones at the same time. Indeed, normal regression predicts a single value for each sample, while this approach allows making several estimations and predicting Y numerical values from an example given as input; in our case, Y is the number of grid cells [20].

Since multi-output regression supports the output of multiple variables for each prediction, adapted machine learning algorithms are needed. We will use classical machine learning algorithms with reduced complexity in order to make accurate predictions for each period within the cells of the grid. In this context, we trained several algorithms and tested them to identify the most suitable one for our specific task, mainly:

- **Multi-Output Regressor:** The Multi-Output Regressor consists of representing each target by exactly one regressor in which its examination may lead to gaining information and predicting future values [21]. In our case, the Multi-Output Regressor algorithm is used in combination with the Stacking Regressor [20] or the Gradient Boosting Regressor [21] since they do not support multiple output regression.
- **Stacking Regressor:** The Stacking algorithm integrates the forecasts from many estimators such as K-nearest neighbors, using an assembly process. Instead of simply gathering the results to retain the majority forecast, the model asks a final estimator to learn to distinguish between the right and erroneous answers [20], [21].
- **Gradient Boosting Regressor:** The Gradient Boosting Regressor is a special case of the boosting method where errors are minimized by the gradient descent algorithm. It consists of a prediction model in the form of many decision trees in order to reward weak learners and generate a final accurate prediction. Weak learners are parameters that perform slightly better than random choices. Unlike the Random Forest algorithm, decision trees in Gradient Boosting are built additively, one at a time [21].
- **Random Forest Regressor:** A Random Forest algorithm consists of a set of independent decision trees from which it performs parallel learning. Thus, it reduces the prediction variance of a single decision tree to enhance its performance. Each tree in the Random Forest is trained on a different random subset of the data, using the bagging principle. The results are then averaged [20].

A machine learning model consists of several extrinsic and intrinsic parameters which are required by the model when making predictions. The adjustable parameters, or hyperparameters, are tuned; they allow us to control the tuning process. This process involves playing with hyperparameters throughout the learning phase until the ideal set of values is found. Thus, it plays a key role in finding the best model by minimizing the predefined loss function. The process of adjusting hyperparameters when training the model is known as the hyperparameter tuning stage.

4) Model's evaluation: Finding the best model that accurately and precisely predicts scooter demand with a low error rate is the aim of our evaluation step. We measure our model performance by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), computed as follows [22]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\sum_{j=1}^{q}|y_{ij}-\hat{y}_{ij}|}{q}\right) \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{j=1}^{q}(y_{ij}-\hat{y}_{ij})^2}{q}} \tag{2}$$

Where y_i and $\hat{y}_i$ are the vector of actual and predicted outputs for x_i, respectively; q is the number of outputs and n is the number of instances. The evaluation metrics in regression problems enable us to precisely measure the error rate of our models by assessing the average difference between the observations (true values in the test dataset) and model output (predicted demands). These criteria's constant range of values is $[0, +\infty[$. In fact, the closer the value is to 0, the more accurate the model's predictions are.

5) ML-RDO exploitation: Considering the operator's needs and capacity in accordance with the strategy employed, our proposed prediction solution can be used to improve the redistribution operations and the quality of the user experience. Additionally, our tool is adaptable by modifying the size of the spatial units according to the operator's service zone size. In this context, the operator can choose between two strategies: (i) a customer-based strategy, in which he can put in place an incentive plan to ask customers to help in the redistribution

process dynamically and return the fleet in popularly predicted areas where there will be high pick-up demand in the future; (ii) an operator-based strategy, by doing the redistribution operations by workers when necessary, after analyzing the results of the prediction.

B. Enhanced ML-RDO

Due to the temporal dependence of our data, we suggest using time-series techniques based on a more potent approach such as deep learning, which can improve our prediction outcomes. Time-series forecasting is a crucial area of machine learning that is often overlooked due to the time component that makes this technique more challenging to handle [23]. The previously mentioned advanced technique is typically used for event prediction based on time-stamped historical data. It predicts future values based on the complex processing of current and previous data, with the assumption that future patterns will be similar to historical trends. For our case, we fully employ the multivariate forecast since we have multiple outputs using a single-step model that learns to predict the next value of the series based on a given history, known as lookback windows or time steps. One of the powerful algorithms that have excellent predictive ability for time-series data is the LSTM network [10], [11]. It learns a function that maps a series of previous observations as input to an output observation. Based on RNNs, which has a structure that repeats itself, LSTMs can retain longer sequences of data and simulate local and temporal dependencies by using dedicated gates to keep the details of previous hidden states h_{t-1} in memory cells c_t and manage the flow of information. There are three gates in a typical LSTM [10]; the input gate i_t decides whether the input should change the content of the cell c_t; the forget gate f_t decides whether to reset the cell content to 0; the output gate o_t decides whether the cell content should influence the neuron's output. Filtering by the output gate o_t is then performed to finally obtain the output h_t.

$$i_t = \sigma(W_i[h_{t-1}, x_t]) + b_i \tag{3}$$

$$f_t = \sigma(W_f[h_{t-1}, x_t]) + b_f \tag{4}$$

$$o_t = \sigma(W_o[h_{t-1}, x_t]) + b_o \tag{5}$$

$$c_t = f_t * c_{t-1} + i_t * tanh(W_c[h_{t-1}, x_t]) + b_c \tag{6}$$

$$h_t = o_t * tanh(c_t) \tag{7}$$

A T timestamp vector series $x = (x_1 ... x_{t-1})$ serves as the input, while the output vector sequence $h = (h_1 ... h_{t-1})$ serves as the output. The cell then combines the input x_t with the prediction at the previous step h_{t-1} to calculate the output h_t; σ and $tanh$ are the two nonlinear activation functions that help the network learn complex patterns in the data. W and b are the weights and biases learned by the model. LSTM's advanced architecture, which combines at timestamp T, hidden state h_t, cell state c_t, and gates, enables it to effectively process temporal data with long-term dependencies by selectively retaining important information from previous time steps.

IV. A CASE STUDY BASED ON TRIP DATA OF E-SCOOTER SHARING IN CHICAGO

Our objective is to help identify the best fleet parking locations to balance the system and optimize fleet deployment based on demand history. The proposed models are trained to predict the number of future pick-up demands within each grid cell, taking into account the selected meteorological and temporal criteria. To implement the machine learning and deep learning models, Python 3.7 was used. A visualization tool was also created with the aid of the Python Folium module [24]. This section includes a presentation of the used datasets, the hyperparameter tuning process, as well as a comparison of various trained model outputs.

A. Dataset description

The trip dataset used in this case study corresponds to 630,816 electric scooter trips excluding errors, collected in the city of Chicago [25], during the five months of summer/fall 2020. It is provided by Pilot, a program to assess the viability of scooter sharing in Chicago. Each transaction record includes details such as trip ID, scooter check-out/in time, start and end locations, trip duration, and trip distance. After converting to date-time format, the instance's check out/in dates are used to extract temporal criteria such as weekday names, days, and months. The meteorological data, which includes temperature, humidity level, wind speed, pressure, and other variables, is gathered from an open-source database [26]. Figure 3, represents the average demand for each hour of the day over the entire period; the demand begins to rise slightly at 10 a.m and peaks at 5/6 p.m. Hence, the rate of use of e-scooters in Chicago rises during the early evening hours and the night. We can also notice that the demand is higher during the weekends than on weekdays. Thus, the temporal criteria are relevant to our case and have an impact on the number of trips.

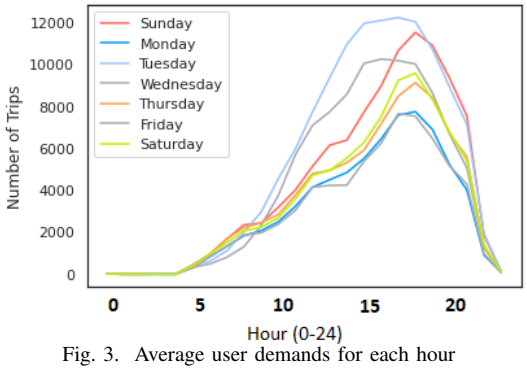

Fig. 3. Average user demands for each hour

In order to conduct a thorough study, we will compare two different grid sizes of 24 and 73 cells. Hence, the spatial unit was set to $\approx 36km^2$ and $\approx 5km^2$, for 24 and 73 grid cells respectively, over $549.077km^2$ Pilot service area in Chicago. Figure 4 and Figure 5, represent a visualization of our grid cells applied to the operational zone[2] after the preparation

[2]In order to use all recorded trips, the cells outside the service area are not eliminated.

phase. In our case, unneeded hexagons, those where no trips were initiated in the whole database, are removed.[3]

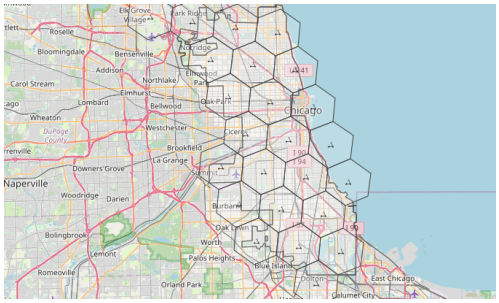

Fig. 4. Hexagon grid visualization (24 cells)

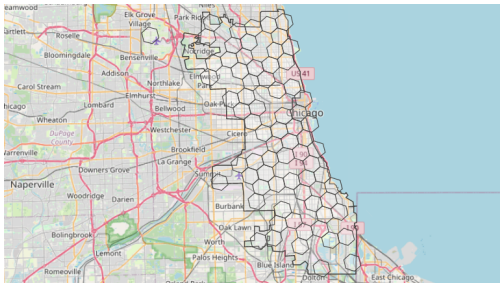

Fig. 5. Hexagon grid visualization (73 cells)

B. ML-RDO hyperparameters tuning

The values of the hyperparameters parameters of the used algorithms, outlined in Section III-A3, are chosen using the random search technique. It enables iterative exploration of all possible set combinations of parameters, with a random way selection to train the model and calculate the score. Table I shows the principal selected hyperparameters of each algorithm.

For Stacking Regressor, the selected estimators are Support Vector Regression, Extra-Trees Regressor, Decision-Tree Regressor, K-nearest neighbors, and Random Forest Regressor as the final estimator of the model [20], [21].

- For SVR, we fix 2 parameters: (i) the parameter C is a regularization parameter that determines how low training error and low testing error are traded off; (ii) the parameter gamma determines the degree to which a single training example has an impact.
- For KNN, we fix the parameter k, which refers to the number of neighbors.

For Gradient Boosting Regressor, we fix:

- Learning rate: it refers to the rate of speed where the gradient moves during gradient descent.
- Loss function: it measures how near the predicted values are to the original label values.
- Max leaf nodes: it refers to the maximum number of leaves in the tree.

For Random Forest, we fix:

- Max depth: it represents the maximum depth of the tree.
- Random state: it controls the bootstrapping's randomness as well as the sample sizes utilized to construct trees.

[3]Due to the lack of unsuccessful demands in the dataset, we were unable to consider them in our analysis. However, it remains conceivable to integrate them if they would be provided.

TABLE I
LIST OF IMPORTANT HYPERPARAMETERS AND THEIR VALUES

Stacking	Gradient Boosting	Random Forest
SVR: C=1e-3 gamma=0.1 KNN: k=5	Learning rate=0.2 max depth=600 loss function=Huber max-leaf nodes=30	max depth=5 random state=0

C. ML-RDO results

In order to forecast the demand within zones throughout the day, our strategy involves applying a prediction by time period, as defined below:

- 6 a.m, 7 a.m, 10 a.m, 11 a.m: Morning
- 8 a.m, 9 a.m: Morning Rush Hour
- From 12 p.m to 3 p.m: Afternoon
- 4 p.m to 6 p.m: Evening Rush Hour
- 6 p.m to 10 p.m: Evening
- 11 p.m to 5 a.m: Night

Table II presents a comparison of the outcomes of the examined models, after the hyperparameters tuning process. The obtained results show that the Gradient Boosting in combination with the Multi-Output Regressor model outperforms all the benchmark models. It also produces efficient and stable prediction results despite the cells' size variation, as indicated by the MAE value of around 0.6 and RMSE values ranging from 0.9 to 1.1. Therefore, the Gradient Boosting algorithm is more adapted to our multiple output problem (i.e. 73 and 24 outputs) than the Random Forest and Stacking models, which are rather suitable for a single output regression.

TABLE II
PERFORMANCE OF ML-RDO MODELS

Metrics	Cells	Stacking	Gradient Boosting	Random Forest
MAE	24-cells	1.946	0.673	1.492
	73-cells	1.079	0.577	0.773
RMSE	24-cells	3.471	1.106	2.604
	73-cells	1.638	0.919	1.165

As shown in Figure 6, we present a visualization of the obtained results for 73 cells on a real map for a selected random instance from the test dataset, mainly the prediction results for a day and a specific time period. In Figure 6, the number of future pick-up demands estimated by the Gradient Boosting in combination with the Multi-Output Regressor model is shown in each hexagonal cell. This demonstrates the precision of our prediction results, as the operator can redistribute the fleets and place them in areas where there will be high user demand in the future. By clicking on the cell, he can know the exact centroid position of the area.

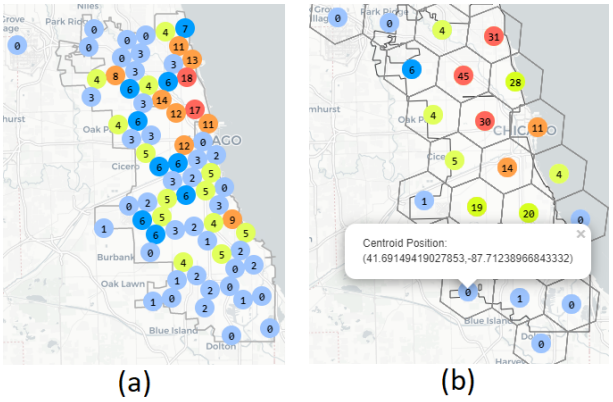

Fig. 6. (a) Prediction result visualization on map (73 cells); (b) Prediction result visualization on map (24 cells)

Figure 7 shows the difference between the predicted and actual values of the previously selected instance. The green color represents the correct values predicted by the model, while the red and yellow colors represent an underestimation and overestimation, respectively. The number in the cells indicates the difference between the actual demands and the predicted demands. Using Eq.1 on this instance, the value of MAE is 0.328 for Figure 7.a and 0.583 for Figure 7.b, which confirms that the predicted user pick-up demand is extremely close to reality. Adapting the redistribution operations according to the results of these predictions improves the availability of the fleets, i.e. the quality of the user's experience.

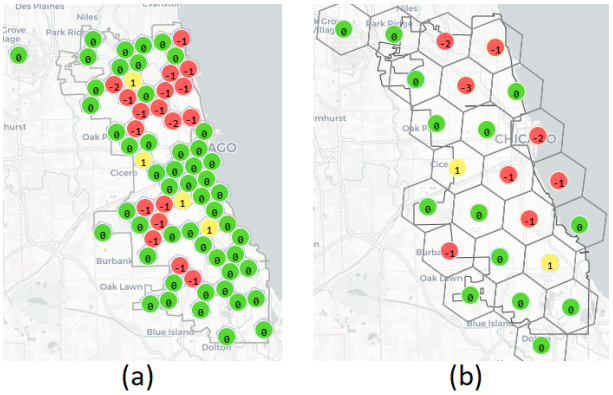

Fig. 7. (a) Visualization of the difference between actual and predicted values (73 cells); (b) Visualization of the difference between actual and predicted values (24 cells)

D. E-ML-RDO results

Since time-series techniques are time-related, we decided to create hourly predictions and estimate the next hour's demand using the previous 12H, 24H, or 48H instances. The database was divided into 80% training and 20% validation data. The weights and biases of the LSTM model were trained using 1e-5 as a learning rate, 50 validation steps, 100 epochs, and Adam [27] as the optimization method, which is ideal for large-scale prediction. To avoid over-fitting, early stopping callback has been used; it enables the training to be automatically stopped when a chosen metric has stopped improving. The Patience parameter of early stopping was fixed at 5; it refers to the number of epochs with no improvement, after which training will be stopped. Table III illustrates the final structure of our model. It combines two layers of neurons, LSTM and Dense layer[4]. The number of trainable parameters represents the number of weights and biases that get updated during the training process in each layer.

TABLE III
LIST OF SELECTED E-ML-RDO'S HYPERPARAMETERS

Layer	Description	Parameters
LSTM (64 neurons)	LSTM input layer	35328
Dense (24/73 neurons)	Deeply connected layer	4680

Table IV represents a comparison of the two proposed solutions using MAE and RMSE as evaluation metrics. With both cell sizes, the enhanced version of ML-RDO shows better performance than the classical one, with an MAE of 0.32 and RMSE of around 0.5.

TABLE IV
COMPARISON OF THE PROPOSED PREDICTION METHODS

Metrics	Cells Number	ML-RDO	E-ML-RDO
MAE	24-cells	0.673	0.322
	73-cells	0.577	0.328
RMSE	24-cells	1.106	0.464
	73-cells	0.919	0.541

Based on different lookback windows of 12, 24, and 48 previous hours, Table V shows the overall evaluation results of the E-ML-RDO solution with both 24 and 73 cells. We observe that despite changing the lookback window, the solution is stable with a slight change using 48 hours compared to 12 and 24 hours, due to the number of previous instances it considers. We can also notice that the enhanced solution outperforms the classical version because of the temporal aspect that has been well exploited by the time-series technique and the LSTM model. This later is able to keep historical temporal information over a long period of time. This is understandable given the significance of the temporal aspect on the e-scooter usage rate in Chicago as shown in Figure 3.

TABLE V
PERFORMANCE OF E-ML-RDO OVER SEVERAL LOOKBACK WINDOWS

Lookback window	Cells Number	MAE	RMSE
12H	24-cells	0.372	0.552
	73-cells	0.376	0.591
24H	24-cells	0.332	0.494
	73-cells	0.331	0.533
48H	24-cells	0.322	0.464
	73-cells	0.328	0.541

Thinking about deploying our solution in a real-time application, this section represents the computing time to make the

[4]A dense layer is used to project the output of the LSTM layer to the desired dimension. The number of its neurons is fixed depending on the cells' number, i.e., 24 or 73.

prediction. Table VI represents the computing time of the ML-RDO solution for the two cell sizes using both the classical solution and the enhanced version.

TABLE VI
ML-RDO COMPUTING TIME

Computing time	Classical ML-RDO	E-ML-RDO
24-cells	0.03470 s	0.02738 s
73-cells	0.03557 s	0.02973 s

We can see that the computing time for both approaches is extremely fast, which demonstrates that our solution is highly suitable for real-time applications. In addition, the enhanced solution slightly outperforms the classic one with $approx$ 28 milliseconds, proving its efficiency.

V. CONCLUSION

In order to solve the maldistribution problem, this study seeks to develop an efficient prediction model based on machine learning algorithms. We evaluated the performance of the proposed ML-RDO approach, using a test case study of the e-scooter system in Chicago. Weather and time variables, such as weekdays, temperature, humidity, and so on, were used to improve our prediction results. Using a hexagonal grid, the Gradient Boosting in combination with the Multi-Output Regressor model is suggested for predicting scooter demand per period, which produces a successful forecast. As a second stage, the LSTM model, as an hourly demand time-series forecasting, provided good results and outperformed the other baselines. This prediction tool enables micro-mobility operators to adapt the launch and redistribution plan to lean to the zones with higher demand, allowing the user to easily find a fleet. Our method has a lower computational complexity with clearer prediction results to facilitate the redistribution process for operators. It is also adaptable to both customer-based and operator-based strategies. Additionally, the proposed tool can be used in a variety of ways such as optimizing pricing strategies, or improving safety while reducing the risk of fleet collisions in popularly predicted areas. In future work, we propose to consider more external features such as traffic volume and events. Additionally, adapting our solution to dock-based systems would be a valuable improvement.

REFERENCES

[1] J. Fong, P. McDermott, and M. Lucchi, "Micro-mobility, e-scooters and implications for higher education," *UPCEA Center for Research and Strategy*, 2019.

[2] S. Shaheen and A. Cohen, "Shared micromoblity policy toolkit: Docked and dockless bike and scooter sharing," *UC Berkeley: Transportation Sustainability Research Center*, 2019.

[3] M. Zakhem and J. Smith-Colin, "Micromobility implementation challenges and opportunities: Analysis of e-scooter parking and high-use corridors," *Transportation Research Part D: Transport and Environment*, vol. 101, p. 103082, 2021.

[4] S. Boglietti, B. Barabino, and G. Maternini, "Survey on e-powered micro personal mobility vehicles: Exploring current issues towards future developments. sustainability 2021, 13, 3692," 2021.

[5] P. Zhou, C. Wang, Y. Yang, and X. Wei, "E-sharing: Data-driven online optimization of parking location placement for dockless electric bike sharing," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2020, pp. 474–484.

[6] Z. Zhang, L. Tan, and W. Jiang, "Free-floating bike-sharing demand prediction with deep learning," *International Journal of Machine Learning and Computing*, vol. 12, no. 2, 2022.

[7] L. Pan, Q. Cai, Z. Fang, P. Tang, and L. Huang, "A deep reinforcement learning framework for rebalancing dockless bike sharing systems," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, 2019, pp. 1393–1400.

[8] M. D. Gleditsch, K. Hagen, H. Andersson, S. J. Bakker, and K. Fagerholt, "A column generation heuristic for the dynamic bicycle rebalancing problem," *European Journal of Operational Research*, 2022.

[9] A. Raptopoulou, S. Basbas, N. Stamatiadis, and A. Nikiforiadis, "A first look at e-scooter users," in *Conference on Sustainable Urban Mobility*. Springer, 2020, pp. 882–891.

[10] Y. Ai, Z. Li, M. Gan, Y. Zhang, D. Yu, W. Chen, and Y. Ju, "A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1665–1677, 2019.

[11] S. Kim, S. Choo, G. Lee, and S. Kim, "Predicting demand for shared e-scooter using community structure and deep learning method," *Sustainability*, vol. 14, no. 5, p. 2564, 2022.

[12] G. Losapio, F. Minutoli, V. Mascardi, and A. Ferrando, "Smart balancing of e-scooter sharing systems via deep reinforcement learning." in *WOA*, 2021, pp. 83–97.

[13] S. W. Ham, J.-H. Cho, S. Park, and D.-K. Kim, "Spatiotemporal demand prediction model for e-scooter sharing services with latent feature and deep learning," *Transportation research record*, vol. 2675, no. 11, pp. 34–43, 2021.

[14] S. Phithakkitnukooon, K. Patanukhom, and M. G. Demissie, "Predicting spatiotemporal demand of dockless e-scooter sharing services with a masked fully convolutional network," *ISPRS International Journal of Geo-Information*, vol. 10, no. 11, p. 773, 2021.

[15] J. Bao, H. Yu, and J. Wu, "Short-term ffbs demand prediction with multi-source data in a hybrid deep learning framework," *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1340–1347, 2019.

[16] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Physical Review A*, vol. 103, no. 3, p. 032430, 2021.

[17] J. Ke, H. Yang, H. Zheng, X. Chen, Y. Jia, P. Gong, and J. Ye, "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4160–4173, 2018.

[18] C. P. Birch, S. P. Oom, and J. A. Beecham, "Rectangular and hexagonal grids used for observation, experiment and simulation in ecology," *Ecological modelling*, vol. 206, no. 3-4, pp. 347–359, 2007.

[19] G. Martin, M. Donain, E. Fromont, T. Guns, L. Roze, and A. Termier, "Prediction-based fleet relocation for free floating car sharing services," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2021, pp. 1187–1191.

[20] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 5, pp. 216–233, 2015.

[21] O. A. Montesinos-López, A. Montesinos-López, J. Crossa, J. Cuevas, J. C. Montesinos-López, Z. S. Gutiérrez, M. Lillemo, J. Philomin, and R. Singh, "A bayesian genomic multi-output regressor stacking model for predicting multi-trait multi-environment plant breeding data," *G3: Genes, Genomes, Genetics*, vol. 9, no. 10, pp. 3381–3393, 2019.

[22] H. Li, W. Zhang, Y. Chen, Y. Guo, G.-Z. Li, and X. Zhu, "A novel multi-target regression framework for time-series prediction of drug efficacy," *Scientific reports*, vol. 7, no. 1, pp. 1–9, 2017.

[23] C. Chatfield, *Time-series forecasting*. Chapman and Hall/CRC, 2000.

[24] Q. Wu, "Leafmap: A python package for interactive mapping and geospatial analysis with minimal coding in a jupyter environment," *Journal of Open Source Software*, vol. 6, no. 63, p. 3414, 2021.

[25] chicago trips data, "Electric scooter trips taken during the 2020 chicago pilot program," accessed 15 february 2023, "http://data.cityofchicago.org/Transportation/E-Scooter-Trips-2020/3rse-fbp6".

[26] weather underground, "Chicago weather history," accessed 01 March 2023, "http://www.wunderground.com/history/daily/us/il/chicago/KMDW".

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.