



HAL
open science

Optimizing Shared Micro-Mobility: Transitioning to Virtual Docking and Demand Prediction for Enhanced System Balance

Rania Swessi, Zeineb El Khalfi, Mohamed Mosbah

► To cite this version:

Rania Swessi, Zeineb El Khalfi, Mohamed Mosbah. Optimizing Shared Micro-Mobility: Transitioning to Virtual Docking and Demand Prediction for Enhanced System Balance. 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC), Sep 2024, Edmonton, Canada. pp.3070-3075, <10.1109/ITSC58415.2024.10920251>. <hal-05242747>

HAL Id: hal-05242747

<https://hal.science/hal-05242747v1>

Submitted on 5 Sep 2025

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.



HAL Authorization

Optimizing Shared Micro-Mobility: Transitioning to Virtual Docking and Demand Prediction for Enhanced System Balance

Rania Swessi¹ and Zeineb El Khalfi² and Mohamed Mosbah³

Abstract—Shared micro-mobility vehicles are perceived as an eco-friendly and cost-effective alternative to traditional public transportation, offering a new urban lifestyle choice. Depending on the strategy employed, micro-mobility companies typically manage their fleets through dock-based or free-floating systems. However, in recent years, the free-floating system has shown its limitations despite its flexibility: inefficiency due to disorganized parking and oversupply. On the other hand, systems based on virtual stations have demonstrated superior efficiency in addressing these drawbacks and incur lower costs than physical stations. This paper proposes a framework for an automated transition from a free-floating system to a virtual dock-based system. This approach uses multi-layer clustering techniques while maintaining high efficiency and service quality, providing demand forecasting within each station for system rebalancing.

Index Terms—Machine learning, multi-layer clustering, shared micro-mobility.

I. INTRODUCTION

Micro-mobility sharing systems have rapidly gained popularity in recent years [1], [2]. With the increasing availability of electric micro-mobility vehicles, companies are purchasing fleets to be deployed in cities, offering consumers the option of renting them via a smartphone application [3]. These vehicles, mainly bikes and e-scooters, have emerged as a trendy and new mobility option in many cities around the world [1], [4]. Besides being lightweight and easily transportable, micro-mobility solutions serve as a simple and enjoyable means of transportation, faster than walking, and perfect for short distances, making them a popular choice for first and last-mile commutes [1], [5], [6]. Additionally, these solutions address key urban transportation issues such as traffic congestion and air pollution, making them ecologically and environmentally sustainable and assisting cities in achieving their social and environmental goals [4], [5].

Two main operational modes are commonly employed to deploy and manage a fleet of micro-mobility vehicles: dock-based and free-floating systems. The dock-based system requires users to pick up and return vehicles at specific physical docking stations. This system offers a secure option for shared micro-mobility vehicles by providing a safe place to store them, with increased protection against damage [7]. Conversely, the free-floating system allows vehicles to be picked up and dropped off freely within an operational area.

Although it is known for its flexibility and ease of use, making it a popular choice among some users, it also has drawbacks such as oversupply and disorganized parking [7], [8], [9]. This has resulted in the occupation of urban places and disruptions to traffic order [9], [10].

In response to these problems, there is a rising preference for virtual dock-based solutions, which utilize digital infrastructure and designated zones. These systems require users to park within specific limits to avoid disorganized parking. These solutions simplify parking, reduce congestion, and offset the significant infrastructure and maintenance expenses associated with physical docking stations. Furthermore, they can be dynamically adjusted to support fleet management and balance, thus maintaining operational and environmental efficiency. However, transitioning to this new system presents challenges, notably in strategically placing virtual stations to manage the fleet and maintain service quality. Rebalancing plays a crucial role as a dynamic operation that helps distribute vehicles and adjust station locations according to user demand. A mismanaged system can lead to a lack of services at key locations, forcing users to travel inconvenient distances to access services [11], [12]. This not only deteriorates the user experience but also affects the system's overall performance, leading to decreased usage rates and reduced company revenue [12]. Therefore, effective rebalancing of vehicles and thoughtful station placement are essential to avoid these issues and ensure the long-term success of virtual dock-based systems.

Although this system is considered innovative compared to others, some studies have suggested integrating virtual stations within a free-floating system [13]. While these studies provide valuable frameworks for integrating virtual stations into free-floating systems, they generally lack detailed precision and do not fully address variations in demand throughout different times of the day [9], [10].

In this paper, we introduce an automated and complete transition from a free-floating system to a virtual dock-based system based on user demand, using multi-layer clustering and density estimation techniques. As a second step, we build on an adaptation of our previously introduced approach [14], which provides precise spatio-temporal demand predictions. We utilize carefully selected temporal and meteorological criteria to enhance accuracy. We further compare the efficacy of the Transformer-encoder model with the established LSTM model presented in [14]. We also discuss how this approach may be deployed to achieve successful rebalancing and station position adjustments. A practical case study utilizing real-world data collected in Bordeaux in collaboration

¹Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, France; Correspondence: rania.swessi@u-bordeaux.fr

²CESI Campus de Bordeaux, LINEACT CESI, France; zelkhalfi@cesi.fr

³Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, France; mohamed.mosbah@u-bordeaux.fr

with Bird [15] demonstrates our approach’s effectiveness and practical applications. The critical contributions of our paper are:

- Development of a multi-layer clustering framework for free-floating system transition to virtual dock-based system.
- Adaptation of the demand prediction tool for system rebalancing and station adjustments.
- An experimental study utilizing a real-world dataset to validate the proposed solution.

The rest of the paper is structured as follows: Section 2 presents a literature review of recent works on free-floating system transition. In Section 3, we describe the proposed approach for virtual station location. Section 4 focuses on system enhancement using demand prediction. Section 5 presents a real case study based on trip data collected in Bordeaux. Section 6 concludes the paper, discussing how the solution can be exploited and outlining future work.

II. LITERATURE REVIEW ON FREE-FLOATING SYSTEM TRANSITION

The primary goal of transitioning to a virtual station system is to reduce the workload due to disorganized parking by relocating poorly parked free-floating vehicles. The transition also reduces environmental costs by eliminating the need for physical installations. For optimal effectiveness, these virtual stations should be strategically located in high-demand areas, such as points of interest. This strategic placement will ensure the system meets user demand and maintains balance. In [13], a prototype of a decision support system and a mathematical model are proposed to help integrate and potentially install virtual stations into a free-floating bike-sharing system. In [8], a graded clustering approach was introduced to identify the locations of virtual stations at different geographical scales in a free-floating bike-sharing system. However, the paper does not provide a precise explanation of how to select the exact position of the station in a cluster or how to manage clusters that are too close. In [10], a mixed linear programming model is proposed for optimizing the location of virtual stations in free-floating systems to maximize user demand during rush hours. In [9], authors use clustering methods to analyze and estimate parking demand in Nanjing by determining the maximum number of bikes within identified clusters. Although their approach provides a static estimation based on observed data, it does not dynamically adapt to fluctuations in demand throughout the day, which is crucial for operational efficiency and effective bike redistribution. Although these methods allow integrating stations efficiently in a free-floating system according to demand, they lack local precision and are limited to specific time intervals, failing to account for hourly variations [9], [10], [13]. Our proposed approach aims to implement an effective solution for a complete and automated transition from a free-floating system to a virtual dock-based system. Additionally, we provide operators with precise hourly predictions using weather and temporal criteria to ensure system quality and balance. Unlike partial transitions,

a complete transition is simpler to manage, more flexible, and dynamically adaptable compared to other operational modes, making it a more efficient solution overall.

III. TRANSITION TO VIRTUAL DOCK-BASED SYSTEM

The main objective of this study is to investigate a complete system transition from a free-floating system to a virtual dock-based system while maintaining optimal service through strategic system rebalancing. This transition necessitates detailed density analysis to accurately capture user demand and strategically position stations for optimal utilization. Our approach fully automates this process, employing a layered clustering method to ensure a thorough, in-depth, and precise analysis of demand patterns. As exposed in Algorithm 1, this method consists of three main steps: data preparation, multi-layer clustering, and determining station positions. The following sections provide detailed descriptions of each step.

Algorithm 1 Clustering-based algorithm for station location

```

1: Input: collected trip data
2: Initialize:  $N, M, minDistance, radius$ 
3:
4:    $\triangleright$  Step 1: Data preparation
5:    $Cdata \leftarrow clean\_and\_preprocess\_data(tripData)$ 
6:    $\triangleright$  Step 2: Multi-layer clustering framework
7:   for  $i \leftarrow 1$  to  $N$  do
8:      $opticsResult \leftarrow perform\_optics(Cdata)$ 
9:      $Cdata \leftarrow opticsResult.Noise$ 
10:  end for
11:  for  $i \leftarrow 1$  to  $M$  do
12:     $hdbscanResult \leftarrow perform\_hdbscan(Cdata)$ 
13:     $Cdata \leftarrow hdbscanResult.Noise$ 
14:  end for
15:   $delete\_noise(opticsResult, hdbscanResult)$ 
16:   $clusters \leftarrow concat(opticsResult, hdbscanResult)$ 
17:   $merge\_nearby\_clusters(clusters, minDistance)$ 
18:   $\triangleright$  Step 3: Station Location
19:   $pos \leftarrow identify\_densest\_points(clusters)$ 
20:   $stations \leftarrow adjust\_points(clusters, pos, radius)$ 
21: Output: list of stations

```

A. Data preparation

Data cleaning and preparation are crucial initial steps before implementing a machine learning model. For our trip data, the process includes deleting or repairing incomplete, duplicate, and damaged data, as well as converting and standardizing the data. We specifically convert feature types into consistent and appropriate forms for our data structure, such as DateTime for timestamps and GPS coordinates for trip locations. These adjustments are crucial for increasing the model’s ability to learn successfully and produce accurate results. We use Python’s GeoPandas and Shapely libraries for

these preparation steps, which are well-suited to handling and interpreting geospatial data.

B. Multi-layer clustering framework

To identify the location of virtual stations, we propose a multi-layer clustering based on the density of user demand across all recorded trip data. By employing multiple layers of clustering, each layer refines the previous one, progressively filtering out noise and honing in on the true demand patterns. This multi-layer approach ensures a precise and homogenous clustering result by processing the data in finer detail at each step. Since we are dealing with geospatial data, we anticipate finding variations within the dataset. To properly address this challenge, we chose two types of algorithms that are specifically designed to handle variation and are well-suited for geospatial data: OPTICS and HDBSCAN [16]. Unlike other clustering methods, they automatically identify clusters based on data patterns and exclude noise without requiring a predefined number of clusters. Their hyperparameters can be adjusted to influence the density and granularity needed to form a cluster.

- **OPTICS clustering algorithm:** The OPTICS (Ordering Points To Identify the Clustering Structure) algorithm is a density-based clustering method that enhances the detection of clusters with varying densities and irregular shapes. It utilizes two main parameters: ϵ , the maximum radius of the neighborhood around a point q , and MinPts, the minimum number of points within this radius to consider q a core point. This radius helps define connectivity, where points beyond ϵ are considered non-reachable, allowing OPTICS to effectively manage large density variations [17], [18].
- **HDBSCAN clustering algorithm:** HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is an advanced clustering algorithm that extends DBSCAN by converting it into a hierarchical clustering algorithm. It is particularly adept at identifying clusters of varying densities. HDBSCAN works by first transforming the space according to the density of the data, then building a hierarchy of clusters and finally condensing this hierarchy based on the stability of clusters [19].

The process is designed to iteratively apply N layers of optics and M layers of HDBSCAN, where the variables N and M depend on the clustering results and noise levels. The more layers we apply, the better we can manage the noise. Tests will be conducted to determine the number of layers necessary to cover all the data. A mix of layers between OPTICS and HDBSCAN allows optimal management of density variations and noise; OPTICS allows detailed exploration of density in data, and HDBSCAN specializes in creating stable clusters. Each layer is composed of micro-mobility trip data, including starting and ending positions over an extended period to capture user demand. The primary layer contains the full dataset, enabling an in-depth examination. Subsequent layers target the noise recovered from the preceding layer, fine-tuning the focus and improving data purity with each

repetition. After combining the insights from all three layers by systematically renumbering and integrating the clusters, we ensure a thorough and coherent examination of the data. The next stage is to validate the clusters and merge those that are too close together, with a precise minimum spacing between their centroids.

To evaluate the effectiveness of our clustering approach, we use several established criteria, each providing insight into distinct aspects of cluster quality.

- **Silhouette Width Criterion (SWC) :** It measures how similar a point is to its own cluster against other clusters. The value is between -1 and 1, where a high value indicates that the point is well suited to its own cluster and poorly suited to neighboring clusters [20]. The silhouette score for a set of samples is given by:

$$SWC = \frac{1}{N} \sum_{i=1}^N \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (1)$$

Where: $a(i)$ is the mean intra-cluster distance. This represents the average distance from the i th point to the other points in the same cluster. Similarly, $b(i)$ stands for the mean nearest-cluster distance. This is defined as the smallest average distance from the i th point to points in a different cluster. N represents the number of clusters.

- **Davies-Bouldin Index (DBI):** This index measures clustering quality by calculating the average ratio of within-cluster scatter to between-cluster separation for each cluster against its closest counterpart. Ideal values of DBI are close to 0, indicating compact, well-separated clusters. The DBI is calculated as:

$$DBI = \frac{1}{N} \sum_{i=1}^N \max_{i \neq j} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right) \quad (2)$$

Where: σ_i is the average intra-cluster distance to the centroid for the cluster i , indicating cluster compactness. $d(c_i, c_j)$ is the distance between centroids of clusters i and j , showing the separation between clusters.

C. Station location identification

To accurately determine the station locations within each cluster, we implement a function that employs the Kernel Density Estimation (KDE) approach. This method identifies the densest point within each merged cluster rather than simply choosing the centroid. It estimates the probability density function of a continuous random variable using sampled data [21]. The density is calculated on a fine grid of points, covering the range of latitudes and longitudes of the points in each cluster. The point with the maximum density on the grid is identified for each cluster as the station's location. This essential step allows us to perform demand predictions correctly and guarantees that we maintain complete trip data.

IV. ENHANCING EFFICIENCY IN VIRTUAL DOCK-BASED SYSTEMS

To increase user satisfaction and improve service quality, it is crucial to constantly balance the system and dynamically

adjust stations according to user demand. This section aims to adapt and test the solution outlined in [14], applying it to our new virtual dock-based system. This solution accurately predicts the number of pickup requests per hour in a free-floating system. It leverages trip data and meteorological and temporal criteria to enhance prediction accuracy. We evaluate this solution within our specific use case and introduce an additional algorithm, comparing its performance with the LSTM prediction model described in [14]. This approach allows the operator to balance and redistribute vehicles or adjust virtual stations dynamically.

A. Approach Adaptation

Due to the uncontrolled parking procedure in the floating system, the approach entails dividing the service area into small spatial units using a cellular grid. Predictions are made simultaneously in each grid cell using multiple output regression. Some preparatory steps are required to adapt this approach to our new virtual dock-based system. These steps involve replacing the grid cells with our list of stations, storing our results in geodata frames, and associating trip data with virtual stations. Other procedures, such as multi-output regression preparation and data fusion, are unchanged.

B. Model's construction

Time series technique effectively tackles hourly prediction challenges by analyzing sequences of time-stamped data [14]. One of the advanced algorithms suited for this task, particularly for handling long-term dependencies and multiple time-based variables, is the Transformer encoder model [22], [23]. It leverages encoding layers to process sequential input data effectively. The Transformer encoder model employs encoding layers to process sequential input data effectively. Key components in adapting this algorithm to our prediction problem include:

- Layer Normalization: Stabilizes the learning process and facilitates faster convergence during model training.
- Multi-Head Attention Mechanism: Enables the model to concurrently focus on different segments of the input sequence, capturing intricate dependencies across various parts of the data.

After passing through the attention mechanism, the input is sent through a dense network, refining the learned representations. This process enables the model to identify deeper connections within the data. Finally, the processed data is fed into a dense layer that predicts the demand at each station for each time slot, effectively addressing the multivariate output requirements of our task.

C. Model's evaluation

The evaluation metrics in regression problems enable us to simplify the process and measure the accuracy of our models by calculating the distance between the predicted and the true actual value across the entire database. We will focus on the following metrics:

- Mean Squared Error (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=1}^q (y_{pred_{i,j}} - y_{true_{i,j}})^2}{q}, \quad (3)$$

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{\sum_{j=1}^q (y_{pred_{i,j}} - y_{true_{i,j}})^2}{q}}, \quad (4)$$

where y_{true_i} and y_{pred_i} are the vectors of actual and predicted values, respectively; q is the number of outputs, and n is the number of instances. The range of possible values for these criteria is $[0, +\infty[$. In fact, the closer the value is to 0, the more likely it is that our model correctly anticipated the outcome.

V. A REAL CASE STUDY BASED ON TRIP DATA FOR SHARED MICRO-MOBILITY IN BORDEAUX

Our approach enables the automatic installation of virtual stations in optimal locations, followed by system adjustments and rebalancing through demand prediction. This allows the operator to benefit from the advantages of a virtual dock-based system while ensuring quality of service. The models are trained on a real-world trip dataset to validate the proposed solution, incorporating selected meteorological and temporal criteria. The Python Folium leaflet package [24] was utilized for result visualization.

A. Data description

We are pleased to collaborate with Bird [15], a micro-mobility operator, to work on their recent dataset on self-service fleet usage. This dataset offers a unique opportunity to study real-world mobility trends in Bordeaux, France, and address imbalance problems. However, the dataset is not yet publicly available due to privacy concerns. The two datasets used in this study are the trajectory dataset and the weather dataset, which are detailed as follows:

- Trip dataset: Bird's trip dataset was collected in the city of Bordeaux. It includes 71,225 recorded trips between arrival and departure positions over the twelve 2021 months and the three winter months of 2022. Each transaction record includes the following details:
 - Trip ID (Object): refers to the trip identifier.
 - vehicle check-out/in time (object): refers to the date of reservation and the return of the fleet.
 - Start and end position latitudes and longitudes (Float64): refers to the trip's exact GPS start and end positions.

Upon conversion to date-time format, the instance's check-out and check-in dates are utilized to extract temporal criteria, including weekday names, days, months, and years. Upon conversion to date-time format, the instance's check-out and check-in dates are utilized to extract temporal criteria, including weekday names, days, months, and years. The weather dataset, sourced from an open-source database [25], contains information on temperature, wind chill temperature,

dew point, wind direction and speed, pressure, visibility, and precipitation.

B. Experimental Results: Transition to virtual Virtual Dock-based system

To adapt the proposed clustering framework to our dataset, we used two layers of OPTICS clustering and one layer of HDBSCAN clustering. Figure 1 displays the clustering results for each layer, applied to the trip data and shown over the service area of the Bird operator in Bordeaux. The blue points represent the noise, i.e. data points that do not fit well into any cluster. The colored points represent the clusters, such as each color indicating a different group of trips. After each layer, we extract the clusters and leave the noise to be processed in the subsequent layer, thereby progressively reducing the noise and refining the clusters.

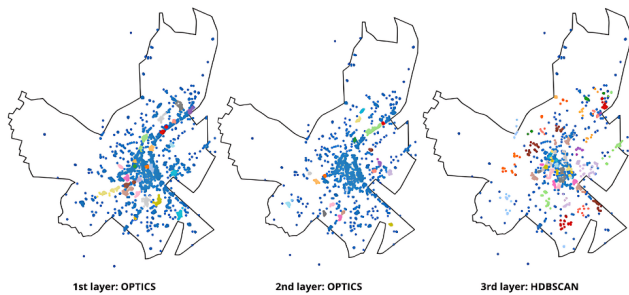


Fig. 1. Layered Clustering Results

The results show that nearly all points are considered and grouped into clusters, resulting in less noise after applying three layers. The OPTICS algorithm focuses on the smaller, denser clusters, while HDBSCAN groups the remaining points based on density and proximity.

Table I illustrates the effectiveness of our clustering framework, evaluated using SWC (eq.1) and DBI (eq.2). Across the three layers, the solution achieves very high SWC scores, averaging 0.9626, indicating robust cluster cohesion and separation. This performance is also confirmed by the DBI, which is close to zero, particularly in the second and third layers, with scores of 0.000825 and 0.001257, respectively. Such low DBI scores affirm that the clusters are well separated and internally homogeneous, emphasizing the fit of the models to the given data and problem, ensuring that the identified clusters are meaningfully distinct and coherent.

TABLE I
CLUSTERING FRAMEWORK PERFORMANCE

Criterion	1st layer	2nd layer	3rd layer
SWC	0.9256	0.9997	0.9997
DBI	0.1368	0.0008	0.0012

For merging nearby clusters, the minimum distance between centroids was set at 300 meters, equivalent to a 3-minute walk. This resulted in 71 clusters in total. Figure 2 illustrates the results of identifying precise station locations using the KDE method after cluster fusion. The analysis

shows that each station is optimally positioned at its cluster’s densest point, including clusters, with relatively few data points. Each station’s radius is set at 15 meters based on operational needs. Because the station does not cover all the trips in its cluster, trips outside the station are placed on the edges of each station to guarantee that we maintain all trip data for the prediction. Figure 3 visually depicts this configuration.

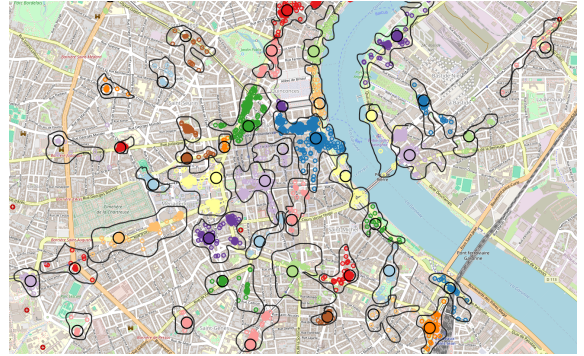


Fig. 2. visualization of station positioning with KDE

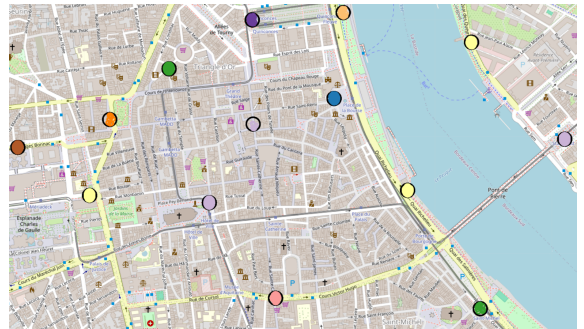


Fig. 3. visualization of stations after adjusting outlying points

C. Experimental Results: Demand forecasting tool

To build a successful model, the hyperparameter tuning phase is crucial [14]. For the proposed Transformer-encoder model, the final values of the hyperparameters are set as follows: the learning rate is fixed at $1e-3$, and the model training is set to run for 100 epochs. Adam is utilized as the optimization method [26]. The architecture of each Transformer encoder block includes two dense layers. The first layer has 128 neurons, and the second layer returns the output to the original input size, matching the transformer model’s input dimension. A dropout layer between these helps avoid overfitting by randomly disregarding some nodes during training. This setup balances complexity and generalization in the model.

Table II presents the outcomes of the adapted prediction tool across all the stations, as detailed in Section IV-B, in comparison with the LSTM model presented in [14]. The results demonstrate that the two models produce successful forecasts with a near-zero error rate. Specifically, the transformer model outperforms the LSTM model, achieving an MSE of 0.2 and an RMSE of 0.4. These metrics represent

the closeness of the predicted values to the actual values, which are very close to zero, indicating the high accuracy and effectiveness of the model. The minimal error margins highlight the Transformer's superior ability to capture and analyze temporal dependencies in the data more effectively than the LSTM model.

TABLE II
PREDICTION TOOL PERFORMANCE

Metric	LSTM	Transformer-encoder
MSE	3.067	0.259
RMSE	0.453	0.445

VI. CONCLUSION

Considering the operator's capacity and requirements within the applied strategy, the proposed solution offers the potential for completely transitioning from a free-floating system to a virtual dock-based system while improving service quality and the overall user experience. The application of multi-layer clustering using OPTICS and HDBSCAN has enabled us to capture user demand density precisely and form stable clusters. By merging these clusters, we avoid the issue of stations being too close to each other. Furthermore, we use the KDE method to determine the exact positions of the stations, ensuring their efficient placement based on the density within these merged clusters. The user demand prediction tool allows the operator to redistribute and balance their fleets or adjust their stations. This process involves relocating fleets or moving stations to areas with high demand for future pickups if the virtual stations become congested during certain periods or simply activating and deactivating stations according to demand. Furthermore, the proposed approach demonstrates adaptability by adjusting solution parameters such as the size of the georeferenced zones in virtual stations and the distance between stations. For future work, we propose automating the dynamic adjustment of the virtual stations created according to demand prediction.

ACKNOWLEDGMENT

This study received financial support from the French government through the France 2030 program IdEx universit e de Bordeaux/RRI ROBSYS and was further facilitated by Bird Global, Inc., which provided us with access to its private data collected on the city of Bordeaux for evaluating the solution on the virtual dock-based system.

REFERENCES

- [1] W. Zhang, X. Niu, G. Zhang, and L. Tian, "Dynamic rebalancing of the free-floating bike-sharing system," *Sustainability*, vol. 14, no. 20, p. 13521, 2022.
- [2] S. Phithakkitnukoon, 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.
- [3] 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.
- [4] J. Fong, P. McDermott, and M. Lucchi, "Micro-mobility, e-scooters and implications for higher education," *UPCEA Center for Research and Strategy: Washington, DC, USA*, 2019.

- [5] R. Chahine, H. Luo, H. Cai, and K. Gkritza, "A comparative study of bike-sharing and e-scooter sharing users and services in a college town during covid-19," *Case Studies on Transport Policy*, p. 101130, 2023.
- [6] T. Campisi, C. Vianello, E. Kuşkan, M. Y. Çodur, and G. De Cet, "Improving the management of operations of the e-scooter services in sicily: A first step of a descriptive statistical survey," *Communications - Scientific letters of the University of Zilina*, vol. 25, no. 4, 2023.
- [7] S. Shaheen and A. Cohen, "Shared micromobility policy toolkit: Docked and dockless bike and scooter sharing," *UC Berkeley: Transportation Sustainability Research Center*, 2019.
- [8] Y. Liu and L. Tian, "A graded cluster system to mine virtual stations in free-floating bike-sharing system on multi-scale geographic view," *Journal of Cleaner Production*, vol. 281, p. 124692, 2021.
- [9] M. Hua, X. Chen, S. Zheng, L. Cheng, and J. Chen, "Estimating the parking demand of free-floating bike sharing: A journey-data-based study of nanjing, china," *Journal of Cleaner Production*, vol. 244, p. 118764, 2020.
- [10] Z. Sun, Y. Li, Y. Zuo, *et al.*, "Optimizing the location of virtual stations in free-floating bike-sharing systems with the user demand during morning and evening rush hours," *Journal of Advanced Transportation*, vol. 2019, 2019.
- [11] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.
- [12] 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.
- [13] C. Rojas, R. Linfati, R. F. Scherer, and L. Pradenas, "Using geopandas for locating virtual stations in a free-floating bike sharing system," *Heliyon*, vol. 9, no. 1, 2023.
- [14] R. Swessi, Z. El Khalfi, I. Jemili, and M. Mosbah, "Free-floating micro-mobility smart redistribution using spatio-temporal demand forecasting," in *2023 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2023, pp. 73–80.
- [15] Bird, "Bird operator," retrieved 1 February 2023, "https://www.bird.co/".
- [16] R. Kuila, P. Sengupta, M. Rout, and R. K. Barik, "Density based geospatial clustering: Methods, applications and future directions," in *Proceedings of the 2022 Fourteenth International Conference on Contemporary Computing*, 2022, pp. 404–409.
- [17] M. Z. Rodriguez, C. H. Comin, D. Casanova, O. M. Bruno, D. R. Amancio, L. d. F. Costa, and F. A. Rodrigues, "Clustering algorithms: A comparative approach," *PLoS one*, vol. 14, no. 1, p. e0210236, 2019.
- [18] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.
- [19] M. Strobl, J. Sander, R. J. Campello, and O. Zaiane, "Model-based clustering with hdbscan," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part II*. Springer, 2021, pp. 364–379.
- [20] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [21] J. Heer, "Fast & accurate gaussian kernel density estimation," in *2021 IEEE Visualization Conference (VIS)*. IEEE, 2021, pp. 11–15.
- [22] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.
- [23] S. Ahmed, I. Nielsen, A. Tripathi, S. Siddiqui, G. Rasool, and R. Ramachandran, "Transformers in time-series analysis: A tutorial. arxiv 2022," *arXiv preprint arXiv:2205.01138*, 2022.
- [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] W. Bordeaux, "Weather bordeaux dataset," retrieved 05 July 2022, "https://prevision.meteo.ch/climat/horaire/bordeaux-merignac".
- [26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.