Data driven methods for effective micromobility parking

William Barbour (corresponding author)
Department of Civil and Environmental Engineering
Institute for Software Integrated Systems
Vanderbilt University
william.w.barbour@vanderbilt.edu

Michael Wilbur
Department of Electrical Engineering and Computer Science
Institute for Software Integrated Systems
Vanderbilt University

Ricardo Sandoval
Department of Electrical Engineering and Computer Science
Vanderbilt University

Caleb Van Geffen
Department of Electrical Engineering and Computer Science
Vanderbilt University

Brandon Hall
Department of Electrical and Computer Engineering
Miami University

Abhishek Dubey
Department of Electrical Engineering and Computer Science
Institute for Software Integrated Systems
Vanderbilt University

Daniel B. Work
Department of Civil and Environmental Engineering
Institute for Software Integrated Systems
Vanderbilt University
1025 16th Avenue South
Nashville, TN 37212

Abstract

Proliferation of shared urban mobility devices (SUMDs), particularly dockless e-scooters, has created opportunities for users with efficient, short trips, but raised management challenges for cities and regulators in terms of safety, infrastructure, and parking. There is a need in some high-demand areas for dedicated parking locations for dockless e-scooters and other devices. We propose the use of data generated by SUMD trips for establishing locations of parking facilities and assessing their required capacity and anticipated utilization. The problem objective is: find locations for a given number of parking facilities that maximize the number of trips that could reasonably be ended and parked at these facilities. Posed another way, what is the minimum number and best locations of parking facilities needed to cover a desired portion of trips at these facilities? In order to determine parking locations, areas of high-density trip destination points are found using unsupervised machine learning algorithms. The dwell time of each device is used to estimate the number of devices parked in a location over time and the necessary capacity of the parking facility. The methodology is tested on a dataset of approximately 100,000 e-scooter trips at Vanderbilt University in Nashville, Tennessee, USA. We find DBSCAN to be the most effective algorithm at determining high-performing parking locations. A selection of 19 parking locations, is enough to capture roughly 25% of all trips in the dataset. The vast majority of parking facilities found require a mean capacity of 6 scooters when sized for the 98th percentile observed demand.

1 Introduction

The launch of shared urban mobility devices (SUMDs) within cities has added a new dimension to the transportation sector — opening up a new way in which to move. Docked bicycles have been established in many cities throughout the world and have seen over 158 million rides since 2010 in the U.S. alone, as of January 2019 [1]. *Dockless* bicycle sharing systems were a paradigm shift that solved some issues around availability by eliminating limits on where the devices could be dropped off. Shared e-scooters operate under the same dockless model, and their rise in popularity has been dramatic; they have been referred to as part of the "micromobility revolution" [2]. As these devices spread across the United States and the world, the number of e-scooter trips exceeded the number of docked bicycle trips by 2 million in 2018. Docked bikes accounted for 36.5 million trips, while e-scooters amassed a staggering 38.5 million trips [1]. The management issues associated with dockless micromobility – such as clutter, accessibility, and safety – have only grown. In order to alleviate some of the pressure caused by a chaotic distribution of scooters, cities have begun to create dedicated parking facilities for SUMDs.

The number of scooter companies in Nashville, Tennessee, USA, has grown to seven as of July 2019. The number of devices grew so quickly due in part to a policy that allows companies to deploy additional scooters after a threshold of three average rides per day is met [3]. The number of trips taken on scooters in Nashville has reached over 1.8 million in the course of approximately one year [4]. Due to an influx of electric scooters, Nashville has experienced enormous amounts of scooter clutter on sidewalks, causing a safety hazard to pedestrians and in some cases, violating requirements for sidewalk clearance under the Americans with Disabilities Act (ADA) [5]. Another notable safety issue concerns where scooters are allowed and intended to ride around a city. In many cities, riding scooters on the sidewalk is illegal due to the risk posed to pedestrians. However, in the presence fast-moving vehicular traffic and without dedicated bicycle lanes, scooter riders are understandably reluctant to ride in the roadway.

There is an obvious demand for cities to adopt forms of SUMD parking management, but there needs to be an effective process by which the locations of this parking are determined. Observations and empirical evidence are insufficient in the placement of this infrastructure. In most proposals for SUMD parking, use of these parking locations would not be mandatory; they would be placed conveniently to collect trips ending near high-volume destinations and their use would be encouraged for riders. This calls for a rigorous, data-based solution calculated over months of micromobility trips. A large dataset has the potential to reveal patterns that may not be obvious to observation and effectively provide governments with the tools to make these infrastructure decisions.

1.1 Contributions

To our knowledge, this is the first study in the United States to propose a data-driven solution for locating SUMD parking facilities and evaluating the efficiency and anticipated utilization of parking facilities. It also provides the first empirical results on how many facilities are needed to serve SUMD demand in an urban campus setting and the capacity allocation and anticipated utilization of such parking facilities. This very well could assist cities in using their own micromobility data to place SUMD parking facilities in the most effective areas.

The remainder of this article is organized as follows. First, we summarize recent literature related to our research. Second, we explain in detail the data-driven methodology . Third, the case study dataset is explained. Finally, the results of the study are presented along with discussion. The article is concluded along with an overview of possibilities for future areas of research.

2 Literature review

In this section, we present a review of relevant literature including safety concerns regarding shared scooters, related policy discussion, and relevant data driven research on SUMD parking.

2.1 Safety concerns

Various forms of micromobility, shared e-scooters included, have gained popularity and interest in part due to their efficiency and ease of use for short, inner city trips [6]. The fact that dockless SUMDs can be picked up and dropped off wherever available has, itself, contributed to many of the safety concerns including the lack of helmet use and hazardous sidewalk clutter. During the Portland's scooter pilot project, the Portland Bureau of Transportation estimated that 90% of riders did not wear a helmet [7]. The results of a study done by Austin Public Health and the Center for Disease Control (CDC) reported that 99% of injured persons were not wearing a helmet when they were injured yet 48% of all injuries observed occured on the head [8]. In a study from Santa Monica, 21 hospital visits concerned non-rider pedestrians, 5 of which were directly related to tripping over scooter clutter in walkways [9]. Better parking allocation could reduce the chances of those types of injuries from occurring.

2.2 Policy discussion

Policy-wise, many changes are being proposed regarding the management of these scooters. Service providers in Shanghai plan on implementing fines to customers parked outside of their "electric fence" parking spots for a bike sharing service [10]. In Nashville, companies are fined $10 for each scooter not in a designated geofence, and $25 for every scooter blocking a right-of-way [11]. Fines are even harsher in San Francisco, where scooter companies are fined over $100 for a parking offense [12]. Along with creating parking fines, there is also mention of potentially using mobile parking infrastructure that shifts based upon current trends [13]. This would allow for a flexible system where data could suggest new locations and the parking infrastructure could be shifted with minimal effort. Improving the scooter infrastructure as a whole is a way to encourage better parking behavior and cause fewer injuries.

2.3 Related micromobility studies

The study on dockless bikes in Shanghai, China presents an optimization framework that is used for the planning of parking infrastructure [10]. The researchers use their optimization algorithm along with the clustering algorithm DBSCAN to determine the optimal locations based on density. Their infrastructure plan only included geofenced parking zones with no physical boundaries. In a study on the spatiotemporal behavior of bicycles, existing docked bike stations in Ningbo City, China, were analyzed using three different clustering algorithms [14]. After performing clustering on all 477 of these bicycle docking stations, they decided that the algorithm k-means had the highest performance and presented their findings with this result. Their objective was to group the stations according to similar characteristics, not according to parking feasibility. A study done comparing the usage patterns of scooters to dockless bicycles discusses the spatiotemporal patterns of bikeshare, but with the added element of comparing it to dockless scooters within Washington, D.C. [15]. Using four months of scooter data from Lime, the patterns found were contrasted with existing data on Capital Bikeshare. While usage trends for both were identified, there was no discussion for potential infrastructure to support scooters.

Existing ways of solving facility location problems, similar to the problem of placing SUMD parking, include: Optimization as used in bike sharing developed for Lisbon, Portugal [16] or a numerical method (i.e. The uncapacitated facility allocation problem (UFAP)) as used to determine facility location for autonomous vehicles in Pittsburgh [17].

Unlike all these similar studies, our study focuses on the parking of e-scooters by combining the use of clustering algorithms to determine facility locations with careful evaluation of these parking locations through temporal capacity analysis.

3 Methodology

In this section we briefly describe our approach to placement and sizing of parking facilities for shared urban mobility devices (SUMDs). First, we outline the use of unsupervised machine learning algorithms to generate high performing parking facility locations. We then discuss details about hyperparameter tuning for relevant algorithms, followed by our process for determining the required size of each parking facility.

By decoupling the generation of parking facility locations and the process for determining the required size of each facility, city officials can determine how to best allocate their available resources. The following methods serve to both provide a comprehensive method for facility deployment, as well as data-driven feedback regarding the SUMD network operations.

3.1 Overview of analysis pipeline

We use three distinct unsupervised machine learning algorithms to first generate clustering allocations of trip endpoints. Each unsupervised machine learning algorithm takes as input a dataset of trip ending locations inside the Vanderbilt campus extents. Exact parking facility locations are calculated using the distribution of points within each cluster, which we explain later.

The performance of a clustering output is determined by the number of trips that fall within a given distance – we refer to this given distance as the *capture radius* – from the exact parking location. This number of trips is referred to as *trip capture*. The objective is to place a particular number of parking facilities such as to maximize trip capture. Multiple runs of each algorithm are performed using different combinations of hyperparameters or different random initializations to achieve the highest trip capture for a specific number of parking clusters (i.e., parking facilities). In this manner, an ideal parking facility allocation is determined at each number of parking clusters, which is then useful in cases where facilities must be prioritized up to a fixed number (e.g., allocating resources for a particular number of facilities).

The relationship between the number of parking facilities and total trip capture works both ways. A desired level of trip capture (e.g. 25% of trips) can also be used to estimate the number of parking facilities required. The number of trips captured per facility, which we refer to as *capture efficiency*, is also an informative metric for decision making. It can help quantify utility or cost-benefit of the infrastructure investment. The following summarize the performance metrics for parking locations:

- *trip capture*: the number of trip endpoints located a distance less than or equal to a given capture radius from the parking location coordinates.
- *capture rate*: the proportion or percentage of trip endpoints captured out of the total number of trips in the dataset; applicable for a single location or totalled across a set of locations.
- *capture efficiency*: the number of trips (or mean number of trips) captured per parking location, given a set of multiple locations.

The dataset used for this work is divided into two mutually exclusive subsets: *training data* and *testing data*. Training data is used to establish parking location with the unsupervised machine learning algorithms. Final performance metrics are calculated and reported on the testing dataset, so as to reduce the potential impact of bias introduced by the training data. The following steps outline the data analysis pipeline used in this work:

1. Collect raw scooter trip data.
2. Preprocess data, apply temporal and geographical filters, and separate data into training and testing sets.
3. Apply unsupervised machine learning algorithms and perform hyperparameter tuning on training data to generate parking clusters.
4. Determine exact parking facility locations based on distributions of trip endpoints within clusters.
5. Assess parking locations by trip capture metric using testing data.
6. Establish parking location capacity requirements using training data.

3.2 Unsupervised machine learning algorithms

We assess the performance of three different clustering algorithms – k-means, DBSCAN, and HDBSCAN – against our objective. Our decision to do so stems from two reasons. First, we want to demonstrate some of the different clustering tools available to decision makers. Second, we want to determine which clustering algorithm has the best performance with respect to our objective, and solves the problem in a timely manner. As each algorithm is comprised of its unique hyperparameters, in addition to the number of clusters, we use a heuristic method for determining the optimal parameters in the context of our objective.

3.2.1 *k*-means

This algorithm is one of the simplest and most popular of the unsupervised machine learning clustering algorithms. The only input parameter required is $n\_clusters$, which allows the user to specify how many clusters in which to partition the data [18]. The k-means algorithm initializes random points as centroids to represent the specified amount of clusters. Then, the algorithm begins a two-step process. First, it assigns all remaining points to its nearest centroid based on euclidean distance (in our case, given 2D space) [19]. Second, it recalculates the centroids by taking the mean of all points assigned to each individual centroid. These two steps iterate until no data points are reassigned or the maximum number of iterations is reached. For more specifics of the k-means algorithm used, reference the documentation in [20].

3.2.2 DBSCAN

The DBSCAN algorithm is a density-based clustering algorithm introduced in [21]. Two of the differentiating factors of this clustering algorithm are its ability to discover clusters of arbitrary shape and its ability to find clusters based on their density, as opposed to only being able to find clusters of a predefined shape or including points that are too sparse to be considered part of a cluster. DBSCAN relies on the input of two parameters by the user: $MinPts$ and Eps-neighborhood. The Eps-neighborhood parameter is referred to in this paper as ε. The idea behind both parameters is the following: determine a dense cluster whose points are within a certain distance ε from each other, and contains at least $MinPts$ points. As a result, DBSCAN classifies points within the dataset as part of a cluster matching the parameterized criteria or as noise [21]. DBSCAN determines a new cluster "by starting at an arbitrary point p, and then retrieving all the density-reachable points based on the hyperparameter combination ε and $MinPts$", as stated in [21]. If DBSCAN is not able to retrieve a density-reachable point, it then moves on to the next point in the dataset and repeats the procedure.

In contrast to k-means and other distance-based clustering approaches, DBSCAN does not take the number of clusters as input. Rather, DBSCAN determines the number of clusters meeting the criteria of $MinPts$ and ε. This is useful in ensuring efficient facilities, but poses a challenge when incorporating resource constraints, or a specific number of facilities, when modeling facility locations. In order to address this challenge and offer a representative analysis for different numbers of facilities, we trained DBSCAN on a wide range of hyperparameter combinations – explained later in the paper – and grouped based on the number of clusters found.

3.2.3 HDBSCAN

The HDBSCAN algorithm was introduced in [22] as an extension to the existing DBSCAN algorithm by "converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based in the stability of clusters" [23]. In this context, hierarchical refers to building a hierarchy of clusters, combining similar points into a cluster at each iteration. Its goal is to find the most significant clusters in a dataset based upon density, while maintaining an efficient runtime and minimal parameters. The only parameter required is $min\_cluster\_size$, which specifies the minimum size desired for a cluster. An optional parameter, $min\_samples$, is often implemented to tune the clustering to the specific dataset. For more information regarding this algorithm, refer to the documentation in [22].

3.3 Hyperparameter tuning

Hyperparameter tuning is an essential step when working with clustering algorithms. The objective of this step is to find the set of hyperparameters for an algorithm that achieve the desired fit to the training data, but generalize well to testing data. In our case, we used hyperparameter tuning to maximize the capture rate of each clustering algorithm at each specific number of clusters. We experiment with hyperparameters on training data in order to get the best parking locations, then assess these parking locations on the testing data.

k-means does not require hyperparameter tuning, but due to its random initialization of centroids, requires numerous runs to determine its best performance. A specific value for $n\_clusters$, is given from 1 to 100 clusters. The algorithm is run with ten initialization and the best result is taken for that value of $n\_clusters$ according to trip capture. Additional initialization were not run due to low variability between each set of results and computational run-time constraints.

As previously described, DBSCAN relies on two parameters: $MinPts$ and ε. We create a range of 233 different values for ε ranging from 2e-05 to 0.0006, and a range of 29 different values for $MinPts$. The values for $MinPts$ are normalized according to the dimensionality of the input data until specified to the algorithm as a number of points. In total, we assess 6,757 different hyperparameter combinations, from which we chose one for each number of clusters that maximized our objective. Following this notion, we find the hyperparameters 0.00018 for ε and 5.75% of the number of points in the training dataset for $MinPts$, which generates 20 clusters. We use this particular clusters to present results throughout this paper.

HDBSCAN, similar to DBSCAN, has two parameters for tuning: $min\_cluster\_size$ and $min\_samples$. For $min\_cluster\_size$, a set of 55 different values ranging from 0.07% to 1.96% is used. There are 39 different values for $min\_samples$, normalized in the same fashion as $MinPts$ in DBSCAN above. 2,145 combinations were evaluated in the hyperparameter tuning, and the most efficient combination was chosen for comparison to the other algorithms.

3.4 Parking facility location and assessment

The exact location of the parking facility must be determined from the cluster assignment generated by each realization of the various algorithms. We first calculate the median longitudinal and median latitudinal coordinates within each cluster, then place the parking location at this coordinate. A circular buffer zone for each parking location is centered at each parking coordinate corresponding to the capture radius. A capture radius of 100 feet is the baseline for analysis. This is the distance required by the US Green Buildings Council (USGBC) for bicycle parking at buildings seeking LEED certification, which numerous Vanderbilt buildings have achieved [24].

3.5 Capacity allocation and analysis

In order to assess capacity requirements for a given parking location, we use the dwell time dataset, described next, to count the number of scooters parked at every minute across the dataset for every trip located within the facility's given capture radius.

4 Case study dataset

4.1 Description of raw data

The dataset was acquired from the Metropolitan Government of Nashville and Davidson County, Tennessee, and covers two of the scooter companies currently operating in Nashville. The historical data spans all trips from August 31, 2018, to March 21, 2019, occurring within the Vanderbilt campus geofence. Data is formatted according to the Mobility Data Specification (MDS) and includes detailed spatiotemporal data for each trip [25]. In our capacity determination, we remove periods of time in which no scooters are parked, including non-operating times (i.e., at night). This is done to reduce bias of capacity for times when the facility is unused, just as would be done for a commercial parking facility or otherwise.

4.2 Preprocessing steps

Data preprocessing first applies a geofilter for the Vanderbilt University campus, retaining only trips for which the trip origin or destination point fell within the campus extents. A temporal filter also removed trips that ended outside of the scooter operating time range: 8:00 to 22:00, inclusive. This range is based on the notion that scooters are typically removed from the streets after 22:00 for recharging, and are then staged the subsequent day around 4:00 to 8:00 [15]. The 8:00 bound is conservative in order to reduce the inclusion of data for morning staging.

We sorted the dataset chronologically before we separated the full dataset into training and testing subsets for parking location determination and assessment, respectively. The dataset was sorted by trip start time and the first 75% of trips were assigned to the training dataset, leaving the latter 25% of the data for the testing dataset. In this manner, the training data covers the date range of August 31, 2018 to March 03, 2019 containing 71,391 trips, and the testing data covers March 03, 2019 to June 21, 2019 containing 23,797 trips

A dwell time dataset was then derived from the original trip data to identify the location and duration each scooter was parked between trips. We identified each scooter by its unique device ID and tracked it through the trip dataset. When a scooter trip ended, the device ID was used to identify when the next trip started. The dwell time was defined as the time between trips. Therefore each dwell consists of a start time, in which the scooter was first parked, an end time in which the scooter started a new trip, and a latitude and longitude coordinate identifying where the dwell occurred. Dwells are also assumed to end each night, meaning that if the end time is past 22:00, that end time is automatically assigned a value of 22:00.

Careful consideration was taken in ensuring that the starting location of the next trip was the same location as the end location of the previous trip. Each device has a natural accepted error of 100 feet, the maximum tolerated distance between the previous trip end location and start location.

4.3 Exploratory analysis of dataset

Exploratory data analysis for scooter trips at Vanderbilt University reveals that trips are generally short in both distance and duration. Distribution statistics for trip distance, duration, and speed are given in Table 1. The median trip distance was 0.69 miles and the mean was 0.97 miles, due to a tail in the distribution of much longer trips. The median trip duration was 6.05 minutes with a

Table 1 : Summary statistics for scooter trips in dataset.

	Mean	Median	Standard deviation
Distance	0.97 miles	0.69 miles	0.92 miles
Duration	9.71 minutes	6.05 minutes	9.05 minutes
Speed	6.28 mph	6.31 mph	3.17 mph

mean of 9.71 minutes, again due to much longer trips. Trip speeds are roughly normally distributed around a mean speed of 6.28 mph. A significant fraction of trips have extremely low speed below 1.0 mph. Some of these are trips with near-zero distance and some are sight-seeing-type trips with modest distance over very long amounts of time.

Demand by day of week for trips at Vanderbilt University is flat from Monday to Saturday, with a slight decrease on Sunday. These trips on and around campus show uniquely stable demand during weekdays due to student use for transportation to and from campus. This is in contradiction to to the general trend in Nashville where demand is significantly higher on weekends.

5 Results

In this section, we present and discuss results of applying the full methodology on the case study dataset. This begins with the data-driven facility placement with unsupervised machine learning, followed by capacity allocation of individual facilities.

5.1 Facility placement and capture potential

As described in the methodology of this work, we explore three unsupervised machine learning (a.k.a., clustering) algorithms for the placement of SUMD parking facilities – a representative, but not exhaustive, selection. The facility locations generated by each algorithm are evaluated with respect to the trip capture within the testing dataset for a capture radius of 100 feet. Trip endpoints that are located within the capture radius of two parking facilities are counted only once towards the total trips captured. Since hyperparameter tuning of certain algorithms will generate multiple realizations with the same number of clusters but different locations, we report for each number of clusters the maximum trip capture rate across all relevant hyperparameter combinations. Note that hyperparameter tuning (only applicable for DBSCAN and HDBSCAN) is not exhaustive; certain numbers of clusters were found many times by various combinations of hyperparameters, while some numbers of clusters were rarely found. The results for k-means, DBSCAN, and HDBSCAN algorithms are shown in Figure 1.

DBSCAN out-performed HDBSCAN and k-means at most numbers of clusters, in terms of total capture rate. Its performance was more variable at higher numbers of clusters, which could be due in part to fewer hyperparameter combinations generating these higher numbers of clusters. At low numbers of clusters, k-means struggled to generate effective clusters, likely due to its mechanism for determining cluster locations. As k-means is not density based, at low numbers of clusters it finds large, sparsely distributed clusters with non-optimal facility locations. For this

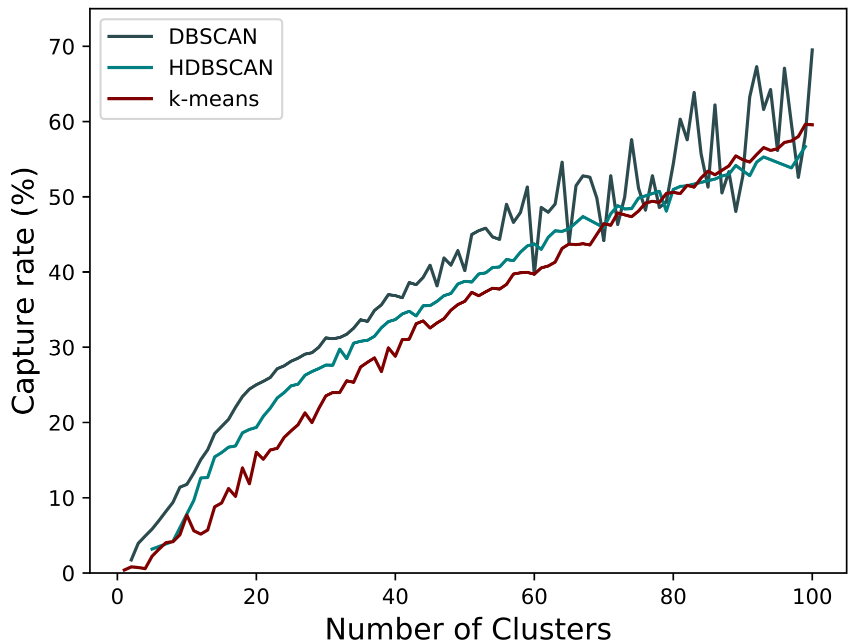


Figure 1 : Comparison of unsupervised machine learning algorithms for the placement of parking facilities, according to number of trips captured within 100 feet.

reason we find density based methods such as DBSCAN and HDBSCAN provide more consistent results regardless of the number of clusters.

During the decision-making process on the number of parking facilities to implement, two metrics will likely be considered, which we have described previously: the total proportion of trips captured across all facilities (capture rate) and the number of trips captured per parking facility (capture efficiency). While the total capture rate generally increases as additional facilities are added, capture efficiency eventually reaches a point of diminishing returns — meaning that as more facilities are added, the capture rate per facility starts to decrease significantly. This knowledge is useful in determining the number of facilities to be installed that would potentially maximize the return on investment. This efficiency tradeoff for DBSCAN is apparent in Figure 2. A hinge point is noticeable in the capture rate curve, where the capture efficiency begins to decrease sharply; this occurs at approximately 19 parking facilities. Selected benchmark values taken from these results are summarized in Table 2.

This relationship between the number of clusters, the capture rate and the capture efficiency can be interpreted as a sort of policy. The total capture rate by a given number of parking facilities provides clear guidance on the expected coverage of SUMD demand in a given area. The capture efficiency metric gives per-facility capture values that are useful to evaluate the return on investment for a set of facilities. For instance, in the case study dataset, a point of rapidly diminishing returns is clearly noticeable, which can help inform a decision on the investment of limited resources. It is important to note that this data-driven procedure is fully consistent regardless of number of facilities chosen or the locale from which the data is sourced. In this sense, outside factors (such as available city resources for deployment) can be taken into account. Data from any city can be used in this same manner.

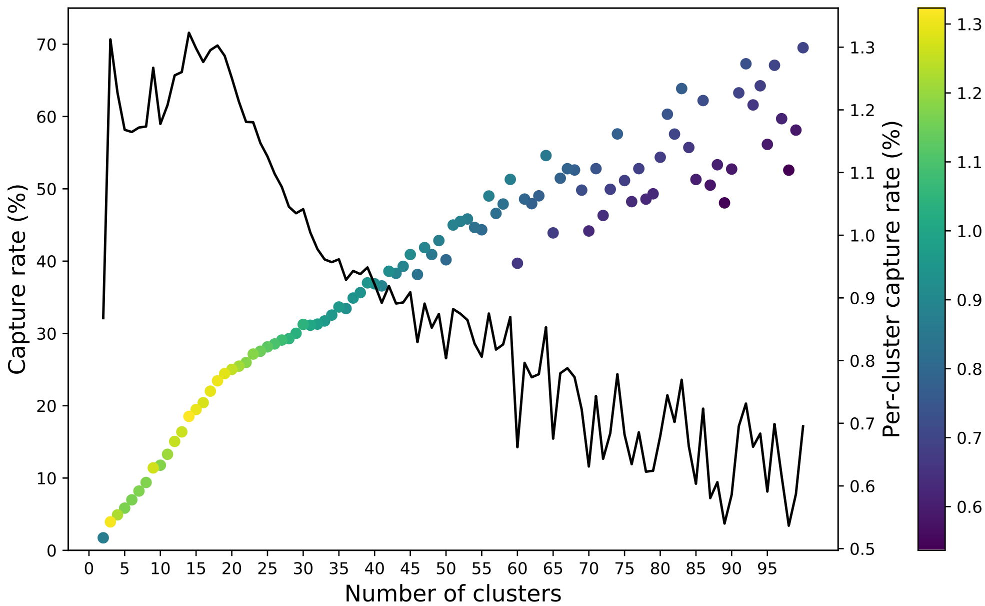


Figure 2 : Relationship between total trip capture (colored points) and per-facility capture rates (i.e., capture efficiency) (black line) for DBSCAN facility placement. *Note: capture efficiency is given on the secondary y-axis.*

Table 2 : Trip capture rate and capture efficiency at various numbers of parking facilities, generated by DBSCAN.

Number of parking facilities	Capture rate (%)	Capture efficiency (% per location)
5	7.0%	1.16%
10	13.3%	1.21%
15	20.4%	1.28%
20	25.5%	1.21%
25	28.5%	1.10%
30	31.1%	1.00%
40	36.6%	0.89%
50	45.0%	0.88%

Table 3 : Parking location names and descriptions

Parking Location	Location Name	Description
A	Rec. Center	Campus recreation and wellness center
B	21st Ave. (North Side)	Adjacent to several restaurants
C	Hillsboro Village	Commercial area for students to eat and shop
D	Rand Hall	Dining hall in the middle of main campus
E	Commons Center	Dining hall in the center of first-year living

The precise locations of parking facilities and their respective capture metrics are dependent on the number of facilities found by or specified to the unsupervised machine learning algorithm, and also by the capture radius. For the remainder of this paper, we present results from a specific parking facility realization generated by the DBSCAN algorithm using the best hyperparameter combination for 19 facilities. This realization was chosen due to its high capture efficiency rate (see Figure 2).

The locations of parking facilities generated under this realization are shown in Figure 3. The cluster assignment of trip endpoints is shown by the color of points (unassigned points are in grey) and a 100-foot capture radius is overlaid on each facility location for scale. The Vanderbilt campus extents are represented by the shaded orange area and Vanderbilt facilities are the hatched outlines.

While there appear to be some areas of concentrated points that are not part of clusters, it should be noted that all of these areas fail to meet the mathematical cluster requirements for this run of DBSCAN. This is an advantage of the algorithm: clear delineation between clusters meeting criteria and all other points.

The capture rates and capture efficiency metrics upon which the algorithms are evaluated depend on the capture radius value that is used for assessment. The results presented on algorithm comparison and capture efficiency dealt with a capture radius of 100 feet [24]. As this value is increased, the number of trips within the capture radius will increase. The rate of this increase is dependent on the density distribution of trips around the facility location. We select a subset of five parking locations (described in Table 3 and shown in Figure 3), labeled A-E from the set of 19 facilities generated by DBSCAN, which are diverse in their location. The effect of increasing capture radius on these facilities can be seen in Figure 4. All location see higher trip capture as capture radius is increased, but location B and C have rapid capture growth between 100 and 200 feet. This indicates that a multi-part parking facility may be advantageous at these locations to capture trips around a larger capture radius.

5.2 Parking facility capacity

We now present parking capacity results for the same subset of five parking facilities A-E. For each location, the number of devices dwelling – according to the last known trip endpoint – within the facility capture radius, is calculated at every minute of the training dataset. We established the capture radius to be 300 feet during capacity evaluation in order to account for extraneous variables, such as GPS inaccuracy. Times at which no scooters are present are removed and the remaining times are arranged in descending order, forming a distribution of the number of scooters

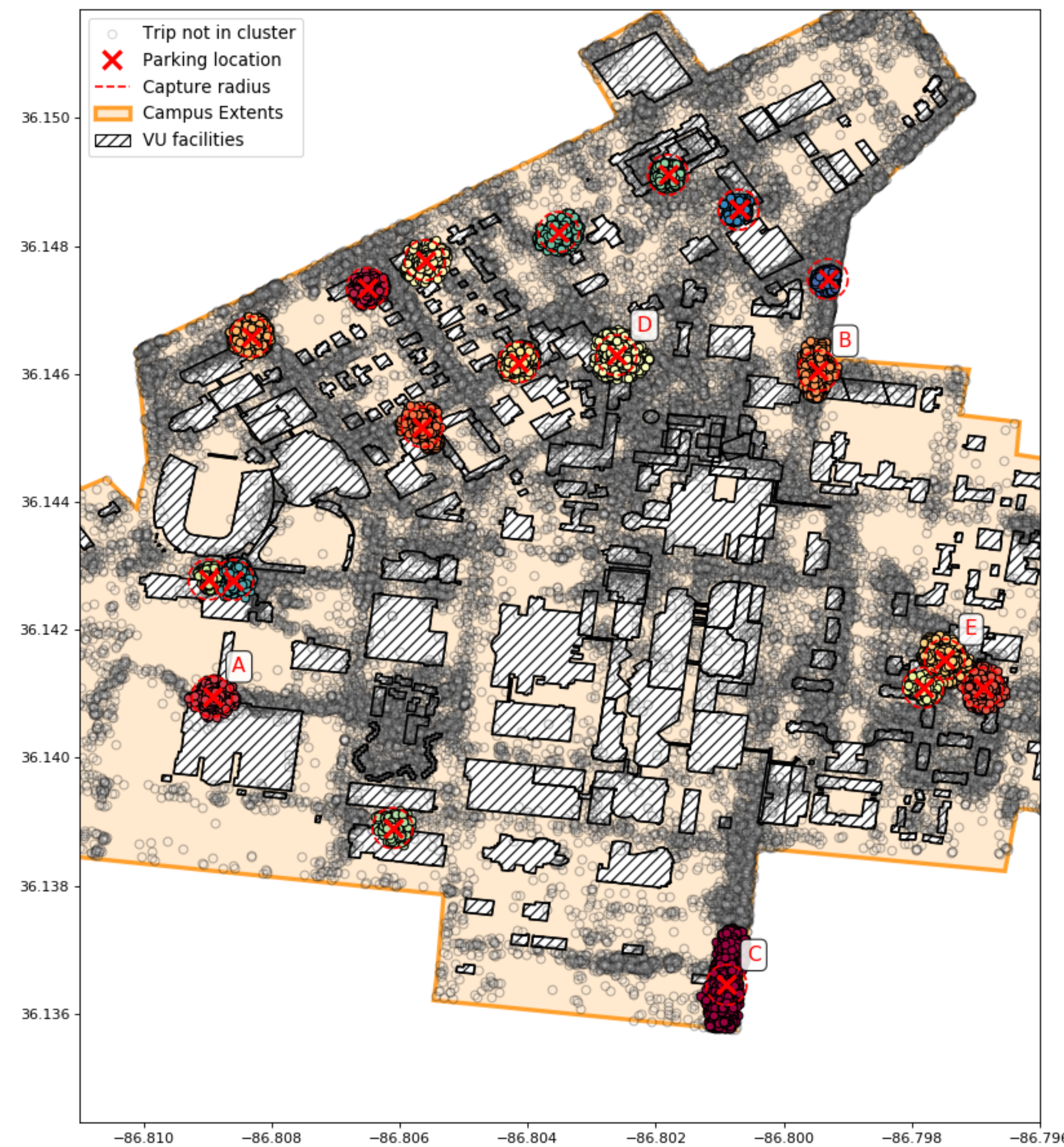


Figure 3 : Map of 19 parking locations (red X's) and capture radii of 100 feet (red dashed lines) generated by DBSCAN. The colored points at each parking location are the trip endpoints belonging to each cluster. Trip endpoints not belonging to a cluster are shown as light grey points. Vanderbilt University campus facilities and campus extents are also shown.

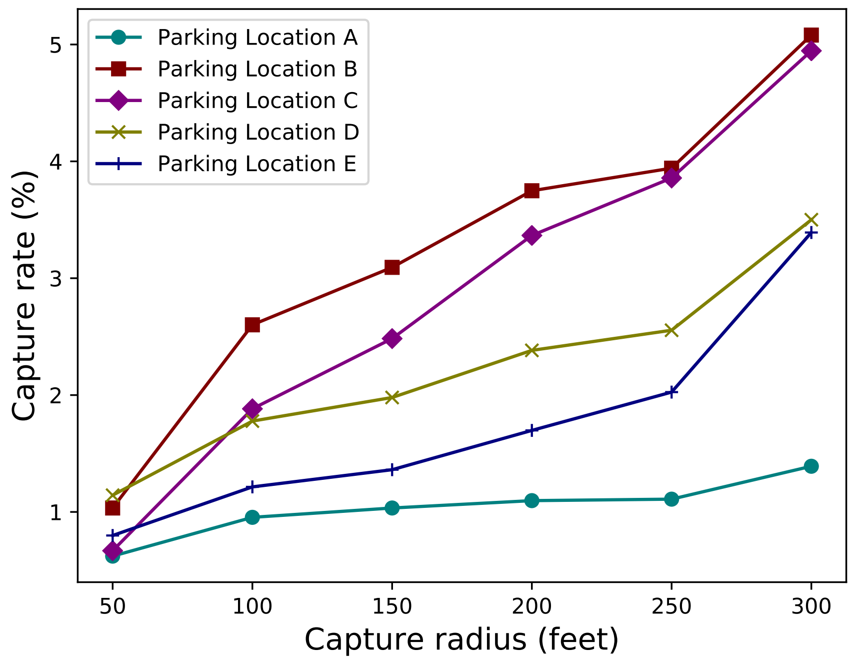


Figure 4 : Growth in trip capture rates as a function of increasing capture radius.

present at a given location at any time. The required capacity for each facility is calculated at specific percentile values for the number of parked scooters: 75, 80, 85, 90, 95, and 98. For example, the 98th percentile capacity value refers to the number of parked scooters that is greater than 98% of all other non-zero counts of parked scooters at a particular location. The distributions of number of parked scooters for locations A-E are shown in Figure 5, and percentile values are shown by colored horizontal lines. As it can be seen in Figure 5 the two parking locations with the highest 98th percentile values are facilities C and D, with a number of scooters corresponding to this percentiles being 7 and 8, respectively. Total area under each distribution is indicative of the amount of scooters parked over time — with locations C and D having a higher area, while location E having a lower area. This indicates that locations C and D have a greater occupancy than location E, because a greater area under the curve is equivalent to a greater occupancy. Additionally, the rate at which the curves decrease also varies across parking locations. While location A has a steep decrease from 12 scooters to 6 scooters in a short period of time, location D shows a lower rate of decrease, taking around 6,000 minutes to decrease from a little more than 13 scooters to 6 scooters. This means that location D is more frequently populated by a larger number of scooters than location A.

These distributions of the number of parked scooters at any given time are sorted and do not take into account the actual time corresponding to the minute that was sampled. Looking at the usage of the selected parking locations by the hour of the day and by the day of the week, as shown in Figure 6, we can see varied trends. Location D (Rand Hall) has much higher utilization during the middle of the day and across weekdays, but falls inline with the other four locations in the evenings and is one of the least-utilized locations during the weekend. Location A (The Rec Center) has uniquely higher demand in the evenings and location B (21st Ave.) has considerably higher demand over the weekend compared to the weekdays.

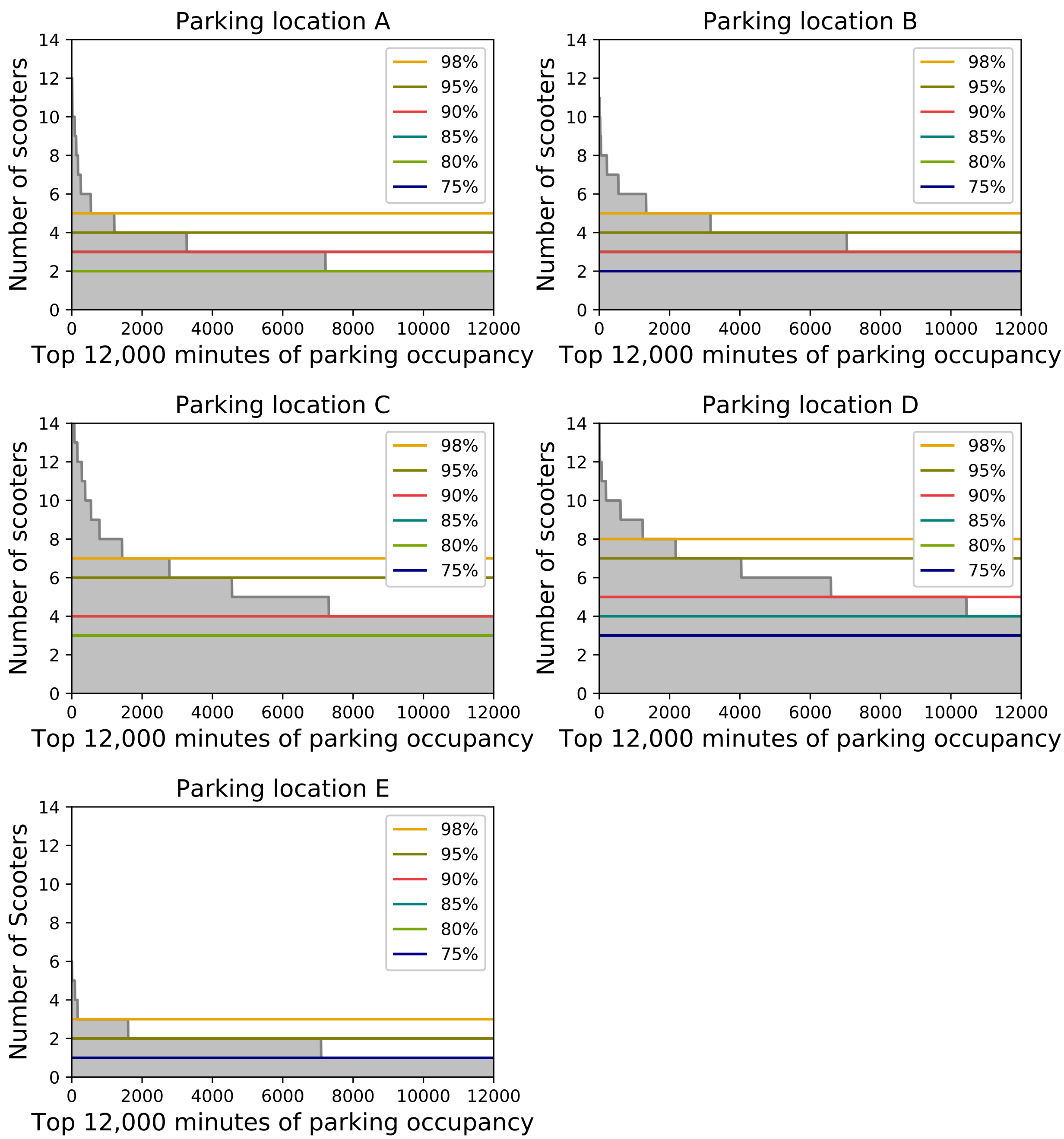
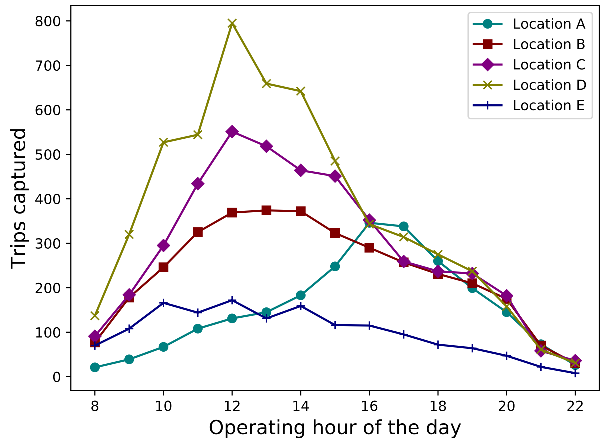


Figure 5 : Distributions of the number of dwelling scooters at specific parking locations, calculated for all times when devices are present. The 75th, 80th, 85th, 90th, 95th, and 98th percentile capacity values are shown for each. *Note that some of these lines may be overlaid on top of each other due to the discrete nature of the capacity values. Also note that the x and y axis have been standardized across the subplots, but Parking Location C has a maximum value of 23 scooters.*



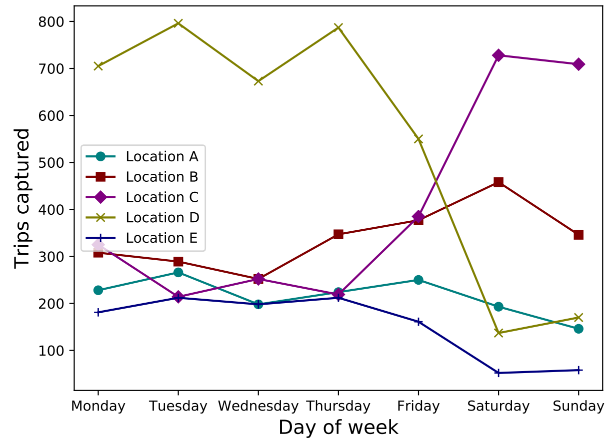


Figure 6 : Distribution of scooter trips captured for parking locations A-E across hours of the operating day (LEFT) and days of the week (RIGHT).

6 Conclusions

This article proposes the data-driven placement of dedicated SUMD parking facilities, so as to maximized their impact in terms of potential capture of trips. These parking facilities are sorely needed in highly congested areas that are prone to clutter of SUMDs, such as e-scooters, on sidewalks. Data-driven placement increases the likelihood that, under an optional usage model, they are effective at convincing users to park devices because of proximity to their destinations. Parking facility locations are determined through the application of unsupervised machine learning algorithms. A case study of these techniques at Vanderbilt University in Nashville, Tennessee, USA, showed that 19 of these locations had the potential to capture 25% of trip demand when evaluated on a previously-unseen testing dataset.

Examination of proposed parking locations over time revealed that the peak capacity required to serve up to 98% of usage periods was likely very low – only 6 devices on average across a set of 19 locations. Clear distinctions were observed in the usage patterns of parking facilities at different locations at Vanderbilt University. Over the course of a day and by day of the week, demand by location varied dramatically. Similar temporal fluctuations would be anticipated in other locales, and the data-driven approach to capacity assessment allows such decisions to consider all time covered by the dataset, instead of anecdotal judgments about location and capacity.

Future work includes accounting for irregularly-shaped (non-circular) clusters of trip endpoints that should probably be served by multiple parking facilities when capture radius does not cover the entire area of dense points sufficiently. Location determination should consider truly feasible facility locations that are free from obstructions such as buildings and roadways. Considering a capture zone that is non-circular in shape due to occlusion by buildings. Furthermore, a model could be created for the likelihood of SUMD users to park in optional dedicated facilities, parameterized by the distance from the facility to their destinations, the area in which the facility is located, and other factors.

7 Acknowledgements

The authors would like to thank the Metropolitan Government of Nashville and Davidson County, Tennessee, (Metro) for their ongoing support. These analyses were derived from information collected by Metro and shared with Vanderbilt as permitted by current legislation through the ongoing partnership between Vanderbilt and Metro.

Financial support was provided in part by the Federal Highway Administration's Office of Innovative Program Delivery (OIPD). This material is based upon work supported by the National Science Foundation under Grant No. CMMI-1351717, and also supported by the National Science Foundation under Grant No. OISE-1743772.

References

[1] Shared Micromobility in the U.S.: 2018. Technical report, National Association of City Transportation Officials, January.

[2] Samantha Raphelson. Dockless Scooters Gain Popularity and Scorn Across the U.S. *NPR*, 2018.

[3] Jeremy Elrod and Burkley Allen. Second Substitute Bill BL2018-1202, 2018.

[4] Jamie McGee. Walk Bike Nashville: Let Scooters Stay, but add Bike Lanes, Fees and Enforcement staff. *Tennessean*, 2019.

[5] 2010 ADA Standards of Accessible Design. Technical report, United States Department of Justice.

[6] Nicole DuPuis, Jason Griess, and Connor Klein. Micromobility in Cities: A History and Policy Overview. *National League of Cities*, 2019.

[7] Portland Bureau of Transportation. 2018 E-Scooter Findings Report. *Portland Bureau of Transportation*.

[8] Dockless Electric Scooter-Related Injuries Study. Technical report, Austin Public Health, April 2019.

[9] Tarak K Trivedi, Charles Liu, Anna Liza M Antonio, Natasha Wheaton, Vanessa Kreger, Anna Yap, David Schriger, and Joann G Elmore. Injuries Associated With Standing Electric Scooter Use. *JAMA Network Open*, 2(1), 2019.

[10] Yongping Zhang, Diao Lin, and Zhifu Mi. Electric Fence Planning for Dockless Bike-Sharing Services. *Journal of Cleaner Production*, 206:383–393, 2019.

[11] Yikyun Jeong. Nashville can fine for illegal parking and operation under new electric scooter rules. *Tennessean*, 2019.

[12] Dan Brekke. S.F. Adopts Tougher Parking Penalties for Scooter Parking Violations. *KQED*, 2018.

[13] Alexandros Nikitas. How to Save Bike-Sharing: An Evidence-based Survival Toolkit for Policy-Makers and Mobility Providers. *Sustainability*, 11(11):3206, 2019.

[14] Xinwei Ma, Ruiming Cao, and Yuchuan Jin. Spatiotemporal clustering analysis of bicycle sharing system with data mining approach. *Information*, 10(5):163, 2019.

[15] Grant McKenzie. Spatiotemporal comparative analysis of scooter-share and bike-share usage patterns in Washington, D.C. *Journal of Transport Geography*, 78:19–28, 2019.

[16] Luis M Martinez, Luís Caetano, Tomás Eiró, and Francisco Cruz. An Optimisation Algorithm to Establish the Location of Stations of a Mixed Fleet Biking System: An Application to the City of Lisbon. *Procedia-Social and Behavioral Sciences*, 54:513–524, 2012.

[17] Andrew Chang, Eunice Choi, Diego Olalde, and Kiran Thapa. Capacitated Facility Allocation with Autonomous Vehicles. *Carnegie Mellon Journal of Mathematical Sciences*, 2016.

[18] David Arthur and Sergei Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proceedings of the eighteenth annual ACM-SIAM Symposium on Discrete algorithms*, page 1027, 2006.

[19] Tapas Kanugo, David Mount, Nathan Netanyahu, Christine Piatko, Ruth Silverman, and Angela Wu. An Efficient k-means Clustering Algorithm: Analysis and Implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[20] M.J.L. de Hoon, S. Imoto, J. Nolan, and S. Miyano. Open source clustering software. *Bioinformatics*, 20(9):1453–1454, 2004.

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[22] Ricardo Campello, Davoud Moulavi, and Joerg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, volume 17, pages 160–172, 2013.

[23] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017.

[24] The US Green Building Council (USGBC). Bicycle facilities.

[25] Mobility Data Specification. Technical report, Los Angeles Department of Transportation, October 2018.