UMAP: Urban Mobility Analysis Platform to Harvest Car Sharing Data

Original
UMAP: Urban Mobility Analysis Platform to Harvest Car Sharing Data / Ciociola, Alessandro; Cocca, Michele; Giordano, Danilo; Mellia, Marco; Morichetta, Andrea; Putina, Andrian; Salutari, Flavia. - ELETTRONICO. - (2017). (Intervento presentato al convegno IEEE Conference on Smart City Innovations tenutosi a San Francisco, California, USA nel August 4 - 8, 2017) [10.1109/UIC-ATC.2017.8397566].

Publisher:
IEEE

Published
DOI:10.1109/UIC-ATC.2017.8397566

UMAP: Urban Mobility Analysis Platform to Harvest Car Sharing Data

Alessandro Ciociola[2], Michele Cocca[2], Danilo Giordano[1], Marco Mellia[1]
Andrea Morichetta[1], Andrian Putina[2], Flavia Salutari[2]
[1]Politecnico di Torino - `first.last@polito.it` [2]Politecnico di Torino - `first.last@studenti.polito.it`

***Abstract*—Car sharing is nowadays a popular means of transport in smart cities. In particular, the free-floating paradigm lets the customers look for available cars, book one, and then start and stop the rental at their will, within a specific area. This is done thanks to a smartphone app, which contacts a web-based backend to exchange information. In this paper we present *UMAP*, a platform to harvest the data freely made available on the web by these backends and to extract driving habits in cities.**

We design *UMAP* with two specific purposes. Firsty *UMAP* fetches data from car sharing platforms in real time. Secondly, it processes the data to extract advanced information about driving patterns and user's habits. To extract information, *UMAP* augments the data available from the car sharing platforms with mapping and direction information fetched from other web platforms. This information is stored in a data lake where historical series are built, and later analyzed using analytics modules easy to design and customize.

We prove the flexibility of *UMAP* by presenting a case of study for the city of Turin. We collect car sharing usage data for over 50 days to characterize both the temporal and spatial properties of rentals, and to characterize customers' habits in using the service, which we contrast with public transportation alternatives. Results provide insights about the driving style and needs, which are useful for smart city planners, and prove the feasibility of our approach.

I. INTRODUCTION

Mobility is one of the challenges to solve in our society and in cities, where eco-sustainability is becoming more and more important. Regulators and policy makers are positively looking into "smart" approaches to improve the current status of their urban network. The ability to collect data, is the first step to take informed decisions. Unfortunately, getting information about mobility patterns and human driving habits is not easy because of both technical challenges and privacy issues.

To this extent, in this paper we investigate the possibility of harvesting data openly exposed on the Web to obtain information about mobility habits in cities, and make it available to the players by using a smart-platform. We focus on car sharing platforms and mapping and direction services.

Car sharing refers to a model of car rental where customers rent a car for a short period of time, usually for a few hours or less. One of its most interesting systems is the so called *Free-Floating Car Sharing (FFCS)* system. The peculiarity of this system is that customers can pick and drop the car wherever in a geo-fence area. The most famous company is car2go which is present in 25 cities and 8 different countries, both in Europe and North America.

To rent a car in a modern FFCS system, users check on their smartphone, or on the FFCS website, which cars are available in the neighborhood. Then, with a simple tap they can book a car, and start/end the rental. The FFCS app contacts a web-based backend server to fetch data about available cars, perform a booking, and accounting operations. Typically for this purpose web API are used, some of which are publicly documented [1]. The same website and app offer information about the status of the car rental systems, and the same web API can be used to collect for free this information. In the past, this approach has been successfully used to obtain data for specific mobility studies – see Sec. II for more details. In this work, we extend this idea and focus our attention on the acquisition and harvesting of this data by means of big data techniques to understand driving habits in a city. We take the city of Turin as a use case.

We design *UMAP*, a platform to collect, process, augment, and store data in a data lake, where analytics let the analyst extract higher level information. We build two crawlers to collect data from the *car2go* and *Enjoy* platforms[1], both present in Turin. Every minute, the crawler checks which cars are currently available. Every time a given car "disappears", it records the booking start time. The same booking ends when the crawler sees the car available back on the system. Some booking are actual "rental" in case the car moved from the prior parking position to another. Ingenuity must be used, e.g., to filter GPS fix issues (which may erroneously let a car "move"), or to handle possible data collection issues (e.g., the website going down, or some cars undergoing in maintenance), or platform design (e.g., synchronous or asynchronous updates).

We let the crawler run to collect data for 52 days, from December 10th 2016 to January 31st 2017. We observed more than 104,000 *bookings* and 86,000 *rentals* for car2go, and 93,000 *bookings* and 81,000 *rentals* for Enjoy. With these datasets, we characterize the FFCS service utilization, in terms of bookings and rentals, with the aim to observe how people use these services, where they typically go, when, for how long the rental last, etc. Some observations are quite intuitive, e.g., people appear to be willing to use more the FFCS during weekdays and during peak-time. Counterintuitively, the rental duration and the driving distance show marginal changes over the day and weeks.

We complement the analysis by comparing the booking duration with the driving duration as suggested by Google Directions application, which we collect in real time for each rental. This allows us to find that 8.5% of bookings last less than the Google driving time. This may be due to Google Directions overestimating the driving duration or, recalling that

---

[1] www.car2go.com, enjoy.eni.com

bookings include the reservation time and the time to look for a parking spot, this may suggest that the time-based tariffs adopted by FFCS systems may encourage fast driving styles in the hope to reduce the rental cost. We next compare the duration of the booking with the equivalent trip duration by public transport as returned again by Google Directions. We discover that rentals are 36% shorter on average than public transport time, but rentals start to be preferred when public transport time is higher than 10 minutes.

We presented our results to the Turin Transportation Authority, who found them to be extremely useful to understand people driving habits. We believe that *UMAP* represents an important support tool for the investigation of car sharing users' habits. The scalable design of *UMAP* allows the policy maker to collect data from many FFCS providers and integrate it with other sources. This eases the analysis when taking in consideration trends and providers comparison. *UMAP* allows the Transportation Authority to take informed decisions when planning public transport systems. This characteristic strengthens the potentiality of *UMAP* for economical and sociological prediction and analysis. Our data-driven approach, combined with other more traditional tools like surveys, represents an interesting observation point for understanding potential services improvements, both for car sharing and public transport systems. We make available the source code of *UMAP* for research purposes.[2]

The reminder of this paper is structured as follows: Sec. II discusses the related work. Sec. III describes in details *UMAP* data acquisition and analysis capabilities. Sec. IV presents our results: First, we characterize car2go and Enjoy car usage over time; second, how customers drive the cars and how they move in the city; finally, we show what are the users' driving habits and the correlation between booking time and the public transport time. Sec. V concludes the paper.

II. RELATED WORK

Since the diffusion of the new form of car sharing based on a free-floating approach, many researchers from different fields have been dedicating an increasing attention to the analysis of these systems. The high demand for car sharing has opened new challenges and perspectives in research.

One of the main topics is the study of fleet relocation policies [2], [3], [4]. On the one hand, with respect to station-based car sharing, the flexibility of the free-floating system may limit the operator's control over the drop-off zones, but on the other hand allows smarter strategies. Herrmann, Schulte and Voß [2] conducted a survey to understand how the availability of cars, and so the fleet relocation, affects the utilization of the service, and to develop and evaluate user-oriented relocation strategies. Those strategies were studied again by Schulte and Voß [3], who introduced an approach to support the decision of vehicle relocation method to reduce costs and emissions in FFCS. Those kind of investigations may result in a very useful support for the providers. In this direction, Wagner, Brandt and Neumann [4], analyzed the use of car sharing in Berlin, using indicators of actractiveness of certain areas, in order to develop a methodology that is able to help in business strategies, the expansion of operative areas and to react to shifts in demand. In these works, the authors used data collected from car sharing providers, using the car2go API [2], [3] or by a direct cooperation [4].

The study of the customers' behavior has been addressed by different researchers [5], [6], [7], [8], [9]. Schmöller et al. [5] studied factors that may influence the demand of car sharing, carrying out an empirical analysis, considering FFCS in Berlin and Munich. Kopp et al. [6] inspected the behavior of two categories of users, the members of a FFCS service (DriveNow), and the people who do not use car sharing (NCS users), looking for different and distinctive mobility patterns. The impact of car sharing on people's mobility was addressed by Firnkorn [7], who proposed in its work a triangulation of two methods applied in the same survey, to provide more precise measurements. Another approach was proposed by Ciari et al. [8], where a simulation tool, built on MATsim, an open source project, was used to estimate travel demand for car sharing in the urban area of Zurich. An important question that can be addressed is how this new paradigm of transport is really accessible to the people. Tyndall [9] combined data of FFCS usage in ten US cites with demographic information, studying neighbourhood infrastructures, population distribution and their mobility habits. It has been showed that benefits of FFCS are distributed unequally, with a shift on usage in favor of advantaged populations.

Eco-sustainability is another important asset for car sharing services. Firnkorn and Müller [10] studied the environmental effects of FFCS in Ulm, registering lower pollution levels and a reduction of private vehicle ownership.

The goal of our work is to address all these challenges from the local administration's perspective, in order to develop new transport and mobility policies. A study of this kind was recently conducted by Wang et al. [11] for the city of Seattle, where car2go was compared with public transport service. Kortum et al. [12] remark the necessity of use data-driven approaches to help decision making, due to the lack of empirical data about free-floating car sharing usage. They use a dataset, obtained by InnoZ (Innovationszentrum für Mobilität und gesellschaftlichen Wandel) and containing the activity in 33 cities from 2011 to November 2015, to study the evolution in time of this mobility service. Those data, combined to demographic informations, offered an aggregated point of view, over different cities, of the growth of the car sharing service and an understanding of the main characteristics. To the best of our knowledge, in the context of our case of study, the only work on free-floating car sharing was conducted by Ferrero et al. [13] from an economical point of view.

The majority of the previous works [2], [3], [4], [5], [6], [11], [12] leverage data collected in real-time or using surveys and interviews. Thanks to car2go APIs, which easily make avaiable car sharing data, a more data-driven approach is attractive for many researchers that start facing the problem of FFCS mobility analysis. Remarkably, only [12] seems to use data collected actively by different car sharing providers. While authors use information only for a specific purpose i.e., analyzing the trend of car sharing through the years, here we want to provide a broader perspective. Our intent is indeed to offer a general purpose methodology, both scalable and easy to interact with, to help researchers and local administrations in the analysis of the mobility, harvesting data collected from FFCS platforms, but also from other online systems, like mapping and direction services.

---

[2] github.com/MobilityPolito/

III. METHODOLOGY

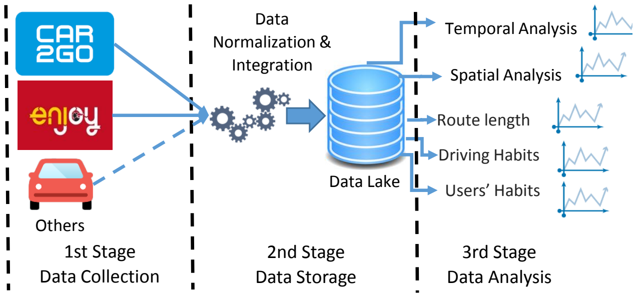


Fig. 1: *UMAP* overview

Our goal is to develop *UMAP*, an integrated system to harvest data freely made available on the web and related to driving habits in cities. *UMAP* offers processing capabilities to perform several analysis and extracting useful information about driving and users' behavior.

In this section, we provide a description of *UMAP*. Figure 1 depicts the architecture of *UMAP*, composed by a first module for the data acquisition, by a second module for data normalization and integration, and then a third module for the data analysis.

A. Data Acquisition

The first module consists in the data acquisition from the car sharing platforms of interest. These typically expose information about cars' location when available for rental through a web-service approach.

For this module we design two crawlers, one for the car2go and one for the Enjoy car sharing platforms. They retrieve, at each time instant, which cars are available in a given city.

While car2go offers public APIs [1], Enjoy does not provide to users such a service. For this reason we study and reverse engineer the Enjoy web portal. By leveraging the Chrome Developer Tools, we investigate the information exchanged with the Enjoy web portal while asking the list of available cars. Through this analysis, we obtain both the URL used to request the list of available cars, and how fetch the data for a specific city. Both system return the currently available cars using a JSON file.

Each time we download a JSON, we discover a new *snapshot* describing which cars are parked and ready for rental.

In a nutshell, a car is described by the car sharing web-service as an object annotated by several information, like plate or vehicle identification number (VIN), location, fuel level, model, etc. All the data represented in this object is useful for the customers e.g., to choose which car to rent. This object is only present if the car is available, i.e., it is parked and free for a rental. Its state changes over time. In particular, a car disappears when a customer reserves and rents it, and then it reappears when the customer ends the rental (likely in a different location).

At each time t, we obtain the JSON snapshot S listing the available cars. The sampling period has been set to one minute, to balance aggressiveness of the crawler and a reasonable time resolution. S describes each available car with several fields, some of them being in common between the considered companies, but in general with different format. For this study, we are interested in each car unique identifier and current geo-location indication. These are obtained from the *VIN* or *plate* field, and the *coordinates* field which describes the *longitude* and the *latitude* of the in-car GPS used to localize it when parked.[3] In addition to these fields, the car sharing JSON description may provide other information, e.g., the *street address* corresponding to the coordinates, the *fuel* level, the *car interior status* the *engine type*, etc. Since each platform uses its own data and format, we design a data integration step to have common names for fields containing the same information, if present.

B. Data Normalization and Integration

In this second module we process and consolidate each snapshot to obtain rentals and parking periods for each car. The intuition is to track the availability of each car on the car sharing platform, and rebuild the historic parking and booking periods over time: when some customer books a car, the latter "disappears" from the system. We record this event, with the initial time and position of a new booking. When the customer ends the booking, the car "reappears" in the system. We record this event, with the final time and position of the booking. For the same car, a new parking period starts.

Harvested data is unstructured, and may grow large. Thus we leverage on *MongoDB*, a NoSQL document-based database. A MongoDB database includes a set collections, i.e., groups of documents. Each document is a set of key-value pairs, organized in a JSON structure. The schema-less structure of MongoDB fits well in our work, because it can handle in the same collection documents defined with different key-value pairs. We decide to rely on such a system as we can easily manage the different field structures of providers, car2go and Enjoy in our use case. In addition, MongoDB offers a great integration with Python through the *pymongo* module.

Four different collections compose our MongoDB data lake: *ActiveBookings*, *ActiveParkings*, *PermanentBookings*, and *PermanentParkings*. *ActiveBookings* and *ActiveParkings* are collections used to store information about the current status of cars (currently booked or parked respectively). These are temporary structures that make it easier to query each car last observed status, and update it. These are also instrumentals for a real-time analysis of the system, e.g., to count how many cars are currently booked or available. *PermanentBookings* and *PermanentParkings* collections store the history of past state of cars, for past bookings and parkings, respectively.

For the documents in the bookings collections we augment information by inserting also the expected route driving time, and the public transportation duration on the same origin-destination pair. These two piece of information are obtained through the Google Directions API using the initial and the final coordinates as indication of the path.

The most important fields in the *ActiveBookings*, and the *PermanentBookings* collections are:

---

[3] The GPS coordinates are only available if a car is parked and available. There is no risk for users' privacy during rentals. In addition no user's identifier is exposed. Therefore data is totally anonymized as there is no means to know who booked a car.

Algorithm 1: Data acquisition at time t

```
   Input : $t$ - Current timestamp
   Input : $S$ - Available Cars (crawling result)
1  $AP = Read(ActiveParkings)$ // Get previous available cars
2  for $car_j$ in $S$ do
3      if ($car_j$ in $AP$) then
4          del $AP[car_j]$;
5      end
6      else
7          $ActiveParkings.add(new\ Parking(car_j, t))$;
8          if ($car_j$ in $ActiveBookings$) then
9              $FinalCoords = car_j[coords]$;
10             $ActiveBooking[car_j][FinalTime] = t$;
11             $InitCoords = ActiveBookings[car_j][InitCoords]$;
12             if (checkCarMovement($InitCoords, FinalCoords$)) then
13                 $ActiveBooking[car_j][driving\_time] = GoogleApi(driving, InitCoords, FinalCoords)$;
14                 $ActiveBooking[car_j][PublicTranportTime] = GoogleApi(public, InitCoords, FinalCoords)$;
15             end
16             $MoveRow(car_j, ActiveBooking, PermanentBooking)$;
17         end
18     end
19 end
20 for $car_j$ in $AP$ do
21     $ActiveParking[car_j][FinalTime] = t$;
22     $MoveRow(car_j, ActiveParking, PermanentParking)$;
23     $ActiveBooking.add(new\ Booking(car_j, t))$;
24 end
```

Fig. 2: Pseudocode of the data acquisition algorithm

- *CarID*: the unique identifier of the car;
- *InitTime*: the initial time of the booking;
- *FinalTime*: the final time of the booking;
- *InitCoords*: the GPS coordinates of the booking star location, i.e., where the users picked up the car;
- *FinalCoords*: the GPS coordinates of the parking location where the car was dropped at the end of the booking;
- *DrivingTime*: The duration of the trip, expressed in seconds, as estimated by Google Directions API, following the best path;
- *PublicTransportTime*: The duration is expressed as arrival time of the best public transport trip, as estimated by Google Directions API, minus the *InitTime*;

Instead, the *ActiveParkings* and the *PermanentParkings* collections are characterized by the following fields:

- *CarID*: the unique identifier of the car
- *InitTime*: the initial time of the parking
- *FinalTime*: the final time of the parking
- *Coordinates*: the GPS coordinates of the parking spot

We implemented an algorithm to extract booking and parking periods from snapshots, whose workflow is described in the pseudocode in Fig. 2. Here we describe each step.

We consider as inputs the snapshot S and the current timestamp t. Then we create a copy in the list AP of parked cars observed in the previous snapshot (as stored in the $ActiveParkings$ collection) – line 1. We need the AP list to detect the cars that disappeared, i.e., have been booked at time t. We will be back on this later.

For each car car_j in the current snapshot S, we check if the car is present in the AP list. If so, it means that it did not change its status, i.e., it is still parked. Therefore, the car is removed from the AP list, and nothing is changed – lines 3-4. Otherwise, either the car has been parked in this snapshot and the previous booking has finished, or the car is a new car added to the fleet. In both cases a new parking starts and we create a new document in the $ActiveParkings$ collection – line 7. The *new Parkings* function creates a new document, sets the $InitTime$ and $Coordinates$ keys as current timestamp and car GPS coordinates.

We next check if car_j is present in the $ActiveBookings$ collection. If so, the car was booked until the previous snapshot and now it is back available. We thus finalize the previous booking and update its statistics. In particular, we set the $FinalCoords$ and $FinalTime$ fields using the current car $coordinates$ and timestamp – line 9-10. Next, we check if this booking includes an actual rental by checking if the initial position and final position differ – line 11-12. Recall indeed that customers may simply book a car but not finalize the rental. Specifically, Enjoy (car2go) offers a grace period of 15 (20) minutes during which no charge is applied for a booking.

In case of an actual rental, we fetch the best path by i) car and ii) public transport from the $InitPosition$ to the $FinalPosition$ of the rental. We leverage the Google Directions API for this – line 13-14.[4] It is important to take into account that, while querying the public transportation time, the Google Directions API returns two pieces of information: how long the public transport takes to go from the initial to the final position, and the estimated arrival time. It is fundamental to use this second information because it includes the time the user spends to reach the bus stop and wait for the bus. This is crucial, e.g., at night, when the first public transport solution may be available only several hours later.

After having processed all cars in the current snapshot, we iterate over the remaining cars in the AP list. Those are the ones that were present in the previous snapshot, but not in the current, i.e., the ones the new bookings. We finalize the previous parking period by setting the $FinalTime$ in the $ActiveParking$ collection – line 21-22. At last, we create a new booking via the *new Booking* function – line 23.

C. Data Analysis

The third and final step is the data analysis phase in which analytics modules query the MongoDB and obtain statistics. We rely on the Python programming language with Pandas and the GeoPandas libraries to deal with the data, the city zone definitions, provided by transport engineers as a shapefile, a popular geospatial vector data format, and the Geographical Information Systems (GIS) for the spatial analyis. We choose Python as it offer a large number of libraries that easily interact with the different technologies like GIS, maps and MongoDB. In particular the usage of GeoPandas allow us to easily perform geographic analysis and split the city in many areas (or zones) of any possible shapes. We describe each analytics in the next section. We present results considering the city of Turin as an example of their usage.

IV. DATA ANALYSIS

In this section we run different analysis to discover and characterize how the FFCS are used. In the first part of the section, we analyze the systems utilization to understand if

---

[4] https://enterprise.google.com/intl/it/maps/products/mapsapi.html

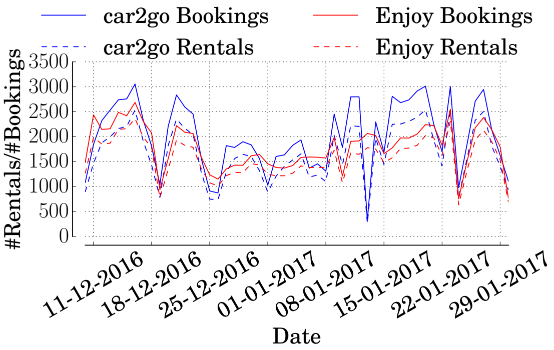


Fig. 3: Total number of bookings and of rentals per day for car2go and for Enjoy

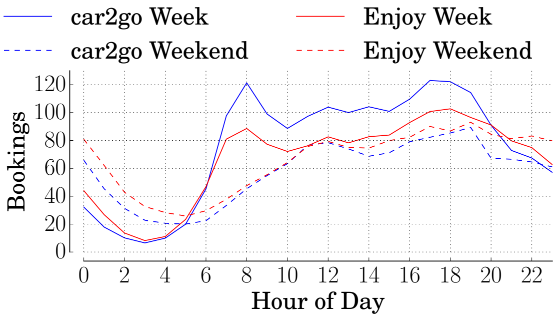


Fig. 4: Average number of bookings during weekdays and weekends for car2go and for Enjoy

FFCS are actually used and when. In the second part, we perform a spatial analysis to analyze how customers tend to move during business days. Finally, we analyze how customers drive FFCS cars and what is the correlation with the public transport system.

We consider a period from December 10th 2016 to January 31st 2017. We observed 125,000 snapshots, about 104,000 bookings for car2go and 93,000 for Enjoy. In Turin, the fleet of car2go was composed by 394 cars, and the fleet of Enjoy was composed by 172 cars.

A. *System Utilization*

Starting from December 10th, Figure 3 plots the total number of bookings and the total number of rentals recorded on each day, for car2go (blue curves), and for Enjoy (red curves). The number of bookings refers to all reservations done by users. Instead, the number of rentals refers only to those bookings where the car final position is different then the initial position. Obviously, being the latter a subset of the first, its number is always smaller. However, during some days, the discrepancy is well visible; that means that the operation of booking cancellation is not so rare.

We can observe both for Enjoy and car2go the same evolution over time, where we can easily recognize a general decrease of number of bookings during the Christmas period, and an increase after the Epiphany. Interestingly, despite the total number of available car2go vehicles is more than twice with respect to Enjoy (394 vs. 172), we can not appreciate such a difference in the number of rentals.

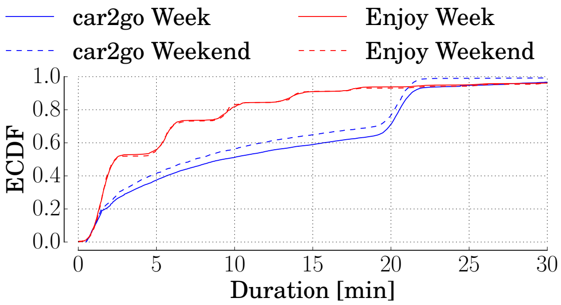
In the figure, some drops in bookings' values are noticeable. Those sudden changes can be addressed to some failures, in the crawlers (e.g., when all curves suddenly drop), or in the operators' web services (e.g., when only one system suffers a sudden drop).

Looking at the data with a finer granularity, we can observe how the car sharing adoption changes during the day. To better characterize this, we separate weekdays and weekends. In Figure 4 we can observe the trend over the day. The curves report the average over the entire period of the number of bookings in each hour of the day. Firstly, we can see that weekdays and weekends have a quite different trend. During the weekend FFCS systems are more used at night with respect than weekdays, with on average at midnight of 80 and 60 bookings per hour for Enjoy and car2go. Instead, we can see how during the weekdays both car2go and Enjoy have their peak of usage at 8 am and between 5 pm and 7 pm. This trend can be easily explained as, during that time slots, FFCS customers use cars to go and return from work. As previously indicated, despite car2go has twice the number of cars than Enjoy, the system utilization of the latter is higher, with peak utilization topping to 60%, versus 30% of car2go. Even in absolute number of rentals, we can see that Enjoy shows an higher number of bookings after 8 pm during the weekdays, and always during the weekends. This can be explained by the car models adopted by the two companies. While car2go uses the compact-two seats *Smart*, Enjoy fleet is composed of *Fiat 500*, which are 4 seats cars. Rentals prices are instead comparable (0.24€/min Enjoy vs 0.25€/min car2go). Data suggests that Enjoy looks more appealing during the times when people prefer to share the ride, and during weekends when families and groups move.

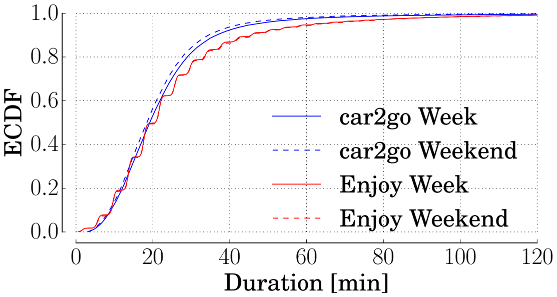
B. *Usage Habits*

Next, we analyze how users tend to use these FFCS systems during weekdays and weekends. We study three different aspects of users behavior: (i) for how long users reserve the car before canceling a booking (Figure 5(a)), (ii) for how long users rent a car (Figure 5(b)), (iii) how far users drive (Figure 5(c)).

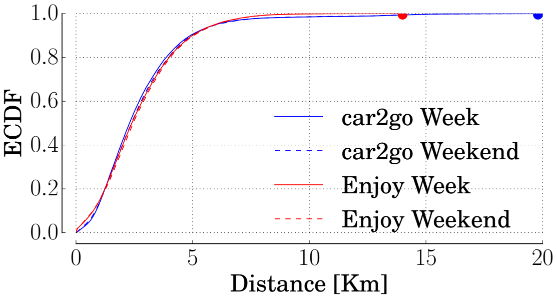
First,we check if and for how long users reserve a car and then they cancel a booking. Interestingly, only a small subset of Enjoy bookings are affected by cancellation with respect to car2go bookings. In particular, in our dataset we observed that 14.9% of all car2go bookings and 2.9% of all Enjoy bookings are cancelled. This again hints for people preferring to use the Fiat 500 offered by Enjoy, so that they hardly cancel a booking when they reserved an available vehicle. On the contrary car2go availability is higher and so it looks easier to find a closer car. People may thus cancel a previous booking when they find a closer vehicle. Another hypothesis is that car2go may be used as a "backup" until an Enjoy vehicle becomes free in the user's area. Looking at when people cancel the reservation, Figure 5(a) shows the CDF of reservation time. We can see how car2go tend to have a smaller percentage of cancellation within 5 minutes, with a huge step at about 20 minutes. While the first ramp can be explained as a communication error or as some sudden cancellation, the latter can be explained by the *maximum free-of-charge reservation time* of car2go. Indeed, users, may reserve a car



(a) ECDF of the booking duration when the booking does not produce a rental. Weekdays and weekends



(b) ECDF of the rental duration. Weekdays and weekends



(c) ECDF of the rental distance. Weekdays and weekends

Fig. 5: Users' booking and rentals habits

up to 20 minutes without paying any fee. Interestingly, we do not see the same trend for Enjoy which offers a *maximum free-of-charge reservation time* of 15 minutes. Instead, we see a peak at 2 minutes and then a decreasing trend after 15 minutes, when almost all the cancellation are already done. One last important aspect that this picture shows is how the Enjoy curves have some steps instead of being smooth as the car2go ones. This hints to periodic updates on web system so that a time granularity emerges. To shed some lights on this phenomenon, we performed some active experiments with the Enjoy web portal. The experiment consists of making a new reservation and find when our crawler detects that the car actually disappears from the set of available cars. Then, as soon as we spot the car disappearing, we cancel the reservation to detect when the car reappears in the system. Surprisingly, we discover that when we make the reservation, the car immediately disappears from the system, instead, when we cancel the reservation, the system takes between 1 and 4 minutes to actually show the car again. The presence of such an offset causes the steps in the Enjoy curves which are affected by an artificial delay. To take into account this offset all Enjoy duration have been decreased of 2.5 minutes, i.e., the average delay the Enjoy system adds.

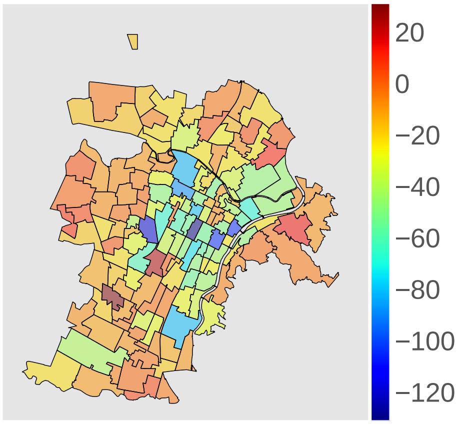


Fig. 6: Heatmap of arrival - departure per area from 7 am to 12 am vs from 5 pm to 9 pm

We next move to characterize the rental duration. We consider only bookings that include an actual rental. Figure 5(b) depicts the Empirical Cumulative Distribution Function (ECDF) of the booking duration for Enjoy and car2go during the weekdays and the weekends. We can see how the trend tends to be equal during the weekdays and the weekends. This demonstrates that despite the different pattern of utilization shown before, the booking duration time is similar. Secondly, we can observe how the ECDF of car2go and Enjoy are almost overlapped, highlighting how these two services tend to be used in a similar way. Most of the rentals last less than 1 hour, with 80% of them lasting less then 30 minutes. It is important to remark that this times include also the reservation time, i.e., the time the user can reserve a car for free before driving it, and the time to find a parking place. Therefore the actual driving time may be significantly smaller.

We repeat the same analysis considering the driving distance as reported in Figure 5(c). To determine the driving distance of each trip we exploit the Google Direction APIs to get the shortest path from the origin to the destination. Similarly for the driving duration, car2go and Enjoy show a comparable behavior, and marginal changes during weekdays and weekends. Interestingly, we see that 90% of the trips last less then 5 km demonstrating that most of the rentals are used for short trips both in term of time, and in term of distance. Lastly, we observe how the car2go curves saturate many km later than the Enjoy ones as highlighted by the circles. This is due to the possibility to reach the airport of Turin with the car2go cars, which is about 20 km far.

C. Spatial Analysis

The present work is designed to offer an easy interaction with geo-spatial tools, and to extract knowledge looking at geographical distribution of cars in the urban area. For this reason we used GeoPandas to consider generic shape zones and correlate them with the FFCS parking places. Figure 6 shows the result as the attractivness of the zones in Turin by analyzing the departure and arrival zones. For each zone we compute the difference between bookings ended in the evening [5 pm - 9 pm] and bookings ended in the morning [7 am, 12 am]. Red areas are those more attractive during the evening, while blue areas are more attractive in the morning. It is clear that the city center is the most popular destination for car sharing during the office hours, while the trips are sparsely ending in the suburbs during the evening.

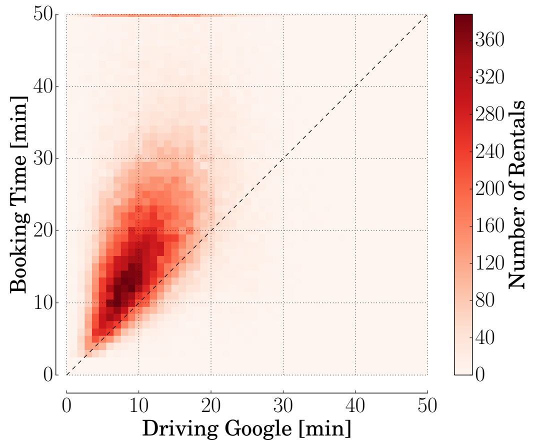
D. Users' Habits

We now characterize how users drive and what is the correlation between public transport usage and availability.

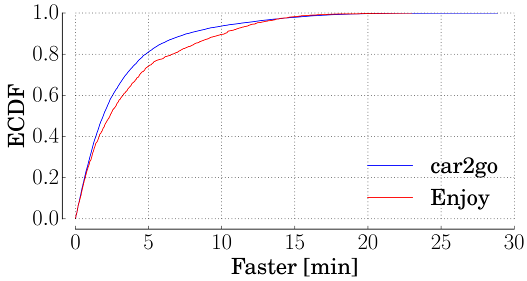
To observe users' driving habits, we use the *driving time* returned by the Google Directions APIs to obtain the estimated driving time from rental initial position to the rental final position. Intuitively, the rental time is longer than the driving time as it takes into account also the reservation time, and the time to find a final parking spot. Figure 7(a) shows an heat map where the X axis represents the Google estimated driving time and the Y axis the actual booking time. Each cell counts the number of observed trips for each (x,y) pairs. For the ease of representation the values are rounded by minute. The diagonal line separates the area where the booking time is lower/greater than the driving time. As we expect, most of the trip falls in the area where the booking time is greater then the driving time. However, a non negligible number of trips (12.1%) falls in the area where the booking time lasts less than the driving time. This may be due to several factors: Google Directions possibly overestimating the average trip duration, or users driving faster than expected. To better quantify how much faster users drive the car in those cases, we compute the difference between the driving time and the actual booking time. We show the Empirical Cumulative Distribution Function of such values in Figure 7(b). As we can see most of these trips are only 5 minutes faster than the estimated driving time, with Enjoy users which seems to drive faster than car2go ones. We verified that in cases where the trip is more than 10 minutes faster, Google suggested a longer path to the destination, e.g., suggesting to take the highway which was much longer with respect to crossing part of the city.

This analysis hints that the current pricing policy, which depends only by the booking time, may have some drawbacks as it may encourage users to drive fast. An hybrid pricing policy, which takes into account both the time and the distance, may be effective in solving this problem, e.g., by increasing the price in case of an user drive faster than expected, or by reducing the fee in case of traffic congestion.

At last, we leverage Google Directions APIs to extract public transport travel information for each vehicle's trip. We want to analyze another way of mobility in the urban area, and compare car sharing usage with respect to public transport. Results are shown in Figure 8. As one could expect, the majority of trips last less than public transport. The higher density is for bookings that last between 10 and 20 minutes. For longer trips, the discrepancy in terms of duration is higher, probably due to the longer path and the higher number of stops of the public transport. Conversely, we can interpret the points where the booking time is greater than the public transport duration as trips where the customers spent a lot of time in reaching the car or finding a parking spot for the drop-off.



(a) Heatmap of booking time vs estimated driving time by Google

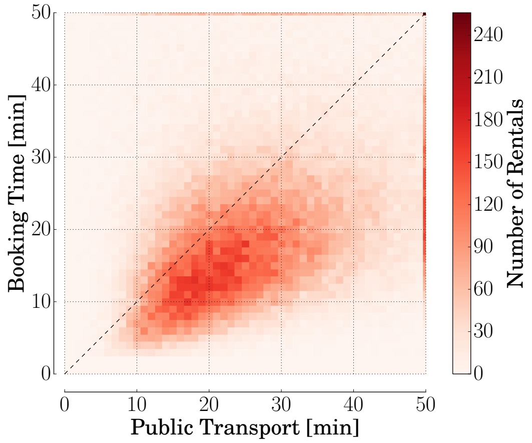


(b) CDF of the difference between the expected driving time and the actual driving time

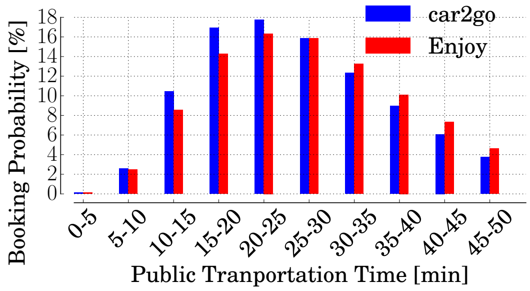
Fig. 7: Users' driving habits

To help to visualize the juxtaposition of car sharing and public transport, we extract from the data the probability of booking a car, conditioned to the public transport travel time. Figure 8(b), reports on the X axis the public transport duration (as predicted by Google) in intervals of 5 minutes, and on the Y axis the probability of booking a car for each interval. The distribution of probability is similar for both car2go and Enjoy. Higher values are reported for trips that can be covered by public transport between 15 and 35 minutes. Interestingly, car sharing mobility is not preferred for very short trips, while the distribution shows a significant tail for duration greater than 30 minutes. This behavior can be justified by the significant amount of time that can be saved with cars haring with respect to public transport.

Finally, to globally understand how users tend to use the different services we report in Figure 9 the average time for: the Enjoy rentals (red curve), the car2go bookings (blue curve), the driving time (green curve), and public transport time (orange curve). To compute this value, for each hour we take all the rentals of interest, and then we compute the average value and report it. A first interesting aspect is that the average time of Enjoy is always greater then the car2go ones and for the pure driving time. Secondly, we can see that both show a similar trend with, a decreasing average duration during the night. As a consequence, we cannot ascribe this trend with traffic jam, instead, but more likely with an increase time in the reservation time and in the parking time. Finally, we can



(a) heatmap of the booking time vs the estimated public tranport time by Google



(b) Probability of using car sharing based on public transport time by Google

Fig. 8: Public transportation vs car sharing

appreciate how during the night the public transport takes more than 1 hour for trips which last less then 20 minutes by car. Instead, we can see how during the daytime the average public transport time get close to the car sharing time.

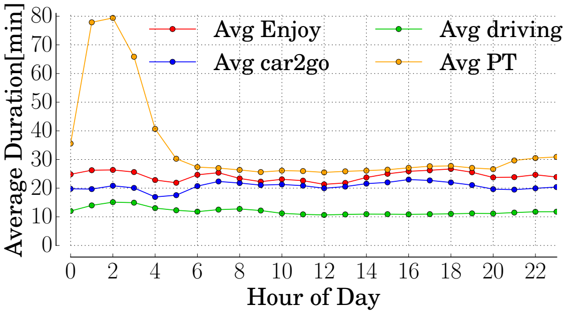


Fig. 9: Average Time per transport solution per hour

V. CONCLUSIONS

In this study we presented *UMAP*, a platform to collect and store data, and able to extract higher level information. By means of two crawlers, we created a 52 days long dataset by collecting data for car2go and Enjoy, two different FFCS operators in the city of Turin. By analyzing the data, we highlighted different aspects related to the system utilization, how users move in the city in different periods of the day, and what are the users' driving habits. This analysis demonstrated how our platform can perform a wide range of analysis in a very simple way. By analyzing the system utilization, we demonstrated that FFCS cars are frequently used for short trips which last less then 30 minutes and 5 kms. We also demonstrated that, despite Enjoy has a smaller fleet, its system utilization is frequently higher than car2go one due to the more appreciated car model it offers. Exploiting the spatial analysis, we highlighted how users tend to move during different time periods. Finally, the users' driving habits showed us that current charging policy may encourage users to drive fast.

The topic is worth further investigation. Thus we invite researchers that are interested to access our dataset and to use *UMAP* that we make them available to the community.

VI. ACKNOWLEDGEMENTS

The research leading to these work was funded by the Siebel Energy Institute.

REFERENCES

[1] "car2go api," https://github.com/car2go/openAPI, accessed: 10/03/2017.

[2] S. Herrmann, F. Schulte, and S. Voß, "Increasing acceptance of free-floating car sharing systems using smart relocation strategies: a survey based study of car2go hamburg," in *International Conference on Computational Logistics*. Springer, 2014, pp. 151–162.

[3] F. Schulte and S. Voß, "Decision support for environmental-friendly vehicle relocations in free-floating car sharing systems: The case of car2go," *Procedia CIRP*, vol. 30, pp. 275–280, 2015.

[4] S. Wagner, T. Brandt, and D. Neumann, "Data analytics in free-floating carsharing: Evidence from the city of berlin," in *2015 48th Hawaii International Conference on System Sciences*, Jan 2015, pp. 897–907.

[5] S. Schmöller, S. Weikl, J. Möller, and K. Bogenberger, "Empirical analysis of free-floating carsharing usage: The munich and berlin case," *Transportation Research Part C: Emerging Technologies*, vol. 56, pp. 34 – 51, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X1500087X

[6] J. Kopp, R. Gerike, and K. Axhausen, "Do sharing people behave differently? an empirical evaluation of the distinctive mobility patterns of free-floating car-sharing members," *Transportation*, vol. 42, no. 3, pp. 449–469, 2015. [Online]. Available: http://EconPapers.repec.org/RePEc:kap:transp:v:42:y:2015:i:3:p:449-469

[7] J. Firnkorn, "Triangulation of two methods measuring the impacts of a free-floating carsharing system in germany," *Transportation Research Part A: Policy and Practice*, vol. 46, no. 10, pp. 1654 – 1672, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965856412001334

[8] F. Ciari, N. Schuessler, and K. W. Axhausen, "Estimation of carsharing demand using an activity-based microsimulation approach: model discussion and some results," *International Journal of Sustainable Transportation*, vol. 7, no. 1, pp. 70–84, 2013.

[9] J. Tyndall, "Where no cars go: Free-floating carshare and inequality of access," *International Journal of Sustainable Transportation*, no. just-accepted, 2016.

[10] J. Firnkorn and M. Müller, "What will be the environmental effects of new free-floating car-sharing systems? the case of car2go in ulm," *Ecological Economics*, vol. 70, no. 8, pp. 1519–1528, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921800911001030

[11] X. Wang, D. MacKenzie, and Z. Cui, "Complement or competitior? comparing car2go and transit travel times, prices, and usage patterns in seattle," Tech. Rep., 2017.

[12] K. Kortum, B. S. R. Schönduwe, and B. Bock, "Free-floating carsharing: City-specific growth rates and success factors," *Transportation Research Procedia*, vol. 19, pp. 328–340, 2016.

[13] F. Ferrero, G. Perboli, S. Musso, and A. G. A. Vesco, "Business models and tariff simulation in car-sharing services," 2016.