Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

An intelligent system for the improvement of the public bicycle rental service

MASTER THESIS

Master Universitario en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Author: Carlos Diez Alba

Tutors: Vicente Julián Inglada
Adriana Giret
Javier Palanca

Course 2016-2017

Abstract

Urban transportation systems have received a special interest in the last few years due to the necessity to reduce congestion, air pollution and acoustic contamination in today's cities. Bike sharing systems have been proposed as an interesting solution to deal with these problems. Nevertheless, shared vehicle schemes also arise problems that must be addressed such as the vehicle distribution along time and across space in the city. Differently to classic approaches, we propose the architecture for a multi-agent system that tries to improve the efficiency of bike sharing systems by introducing user-driven balancing in the loop. The rationale is that of persuading users to slightly deviate from their origins/destinations by providing appropriate arguments and incentives, while optimizing the overall balance of the system. In this paper we present results for one of the system's modules, which will allow us to predict bike demand in different stations. The proposed module has been tested in a realistic scenario, the bike sharing system in the city of Valencia (Spain). Also, a second module is introduced. This module is in charge of calculating and offer to the user the best route an user can travel, using the information generated by the first module, taking into account the balance of the bike-sharing system.

Key words: multi-agent systems; vehicle sharing systems

Resumen

Los sistemas de transporte urbano han recibido un interés especial en los últimos años debido a la necesidad de reducir la congestión del tráfico, la contaminación atmosférica y la contaminación acústica en las ciudades. Los sistemas de bicicletas compartidas han sido propuestos como una interesante solución a estos problemas. Sin embargo, los esquemas de vehículos compartidos también plantean problemas que deben abordarse como puede ser la distribución de los vehículos a lo largo del tiempo y en el espacio en la ciudad. Alejándonos de los enfoques clásicos, proponemos la arquitectura de un sistema multiagente que intenta mejorar la eficiencia de los sistemas de uso compartido de bicicletas introduciendo un balanceo impulsado por el usuario durante el propio uso del servicio. El planteamiento es persuadir a los usuarios para que se desvíen ligeramente de sus orígenes y/o destinos proporcionándoles argumentos e incentivos apropiados, al mismo tiempo que se optimiza el balanceo general del sistema. En este trabajo presentamos los resultados de uno de los módulos del sistema, el cual nos permitirá predecir la demanda de bicicletas en las diferentes estaciones. Este primer módulo ha sido probado en un escenario realista, el servicio de bicicletas compartidas de la ciudad de Valencia (España). Además introduciremos un segundo modulo encargado de calcular y presentar la mejor ruta que un usuario puede tomar, usando la información generada por el primer módulo, teniendo en cuenta el balanceo del sistema de bicis compartidas.

Palabras clave: sistemas multiagente, sistemas de vehículos compartidos

Resum

Els sistemes de transport urbà han rebut un interés especial en els últims anys degut a la necessitat de reduir la congestió del trànsit, la contaminació atmosfèrica i la contaminació acústica en les ciutats. Els sistemes de bicicletes compartides han sigut proposats com una interessant solució a estos problemes. No obstant això, els esquemes de vehicles compartits també plantegen problemes que han d'abordar-se com pot ser la distribució dels vehicles al llarg del temps i en l'espai en la ciutat. Allunyant-nos dels enfocaments clàssics, proposem l'arquitectura d'un sistema multiagent que intenta millorar l'eficiència dels sistemes d'ús compartit de bicicletes introduint un balanceig impulsat per l'usuari durant el propi ús del servici. El plantejament és persuadir els usuaris perquè es desvien lleugerament dels seus orígens i/o destinacions proporcionant-los arguments i incentius apropiats, alhora que s'optimitza el balanceig general del sistema. En aquest treball presentem els resultats d'un dels mòduls del sistema, el qual ens permetrà predir la demanda de bicicletes en les diferents estacions. Este primer mòdul ha sigut provat en un escenari realista, el servici de bicicletes compartides de la ciutat de València (Espanya). A més introduirem un segon module encarregat de calcular i presentar la millor ruta que un usuari pot prendre, usant la informació generada pel primer mòdul, tenint en compte el balanceig del sistema de bicis compartides.

Paraules clau: sistemes multiagent, sistemes de vehícles compartits

Contents

List of Figures

List of tables

CHAPTER 1
Introduction

Transportation systems have become one of the most important areas of application for artificial intelligence paradigms [32, 7, 30, 6]. There are several reasons behind this trend such as the scale of the problem, the need to optimize a pool of limited resources, or the necessity to include models of human behavior in the loop. Among all the available transportation systems, urban transportation systems have received a special interest, boosted by public and government initiatives. With an increasing population in urban areas comes a rise for the need of urban transportation [14]. This rise is problematic as it may lead to problems such as congestion, air pollution, investment into expensive infrastructures, and so forth [14]. In many cases, the investment is two-sided as it also potentially involves citizens acquiring new transportation vehicles for individual use. Hence, optimizing current resources has become an area of great interest for both the public and the private transportation sector.

Shared vehicle schemes, such as bike or car sharing systems, have been proposed as a solution to both optimize the number of existing vehicles, traffic, and as a mean of contributing to a cleaner environment. Despite its advantages, shared vehicle schemes also arise other problems that must be addressed. For instance, one of the problems with bike sharing systems is the bike distribution along time, creating some areas that agglutinate most of the bikes, thus making parking very difficult, and some areas lacking bikes, thus making it very difficult to borrow a bike from that location. In the specific case of bike sharing systems, this can lead to potential dissatisfaction of users, which in the end may result in loss of service subscribers, and an increase in the use of non-shared vehicles like personal cars. Of course, that is usually translated into several problems including traffic jams, rise in pollution problems, or even less healthy citizens due to a more sedentary form of transportation. Moreover, bike sharing providers often need to balance bikes across stations by using trucks or other types of motorized transportation. This incurs in an additional cost for the service provider, as well as more traffic if balance is not done properly.

The problem of optimizing bike sharing systems' resources (i.e., bikes, stations, transportation trucks) has caught the attention of researchers [22, 39, 27, 34], who have proposed many architectures and algorithms that allow service providers to both predict the incoming/outgoing demand from bike sharing stations, as well as educated balancing strategies that optimize the service provider's resources. All of these proposals are pieces of a global strategy that aims to smartly balance bikes according to future demand. All of the actions and strategies are applied from a service provider perspective, while taking the user behavior as granted. This means that resources are optimized by modeling the user behavior, and accepting that behavior as an external effect that will change the system. As a result, actions aiming at balancing the state of the system are solely carried out by the service provider. This work takes a slightly different point of view to this problem. What if, instead of taking the user behavior for granted, we attempt to slightly modify the user's planned trip for optimizing the overall bike sharing system?

The work defines the architecture of a multi-agent system aimed at improving the efficiency of bike sharing system by introducing user-driven balancing in the loop. While predicting the future demand and smartly balancing bikes across stations are seen as important components of the system, we also envision the inclusion of a negotiation and argumentation [31, 10] module that aims to slightly modify the behavior of users. Then, the work moves to analyze and discuss the initial results of one of the core modules of the system: the prediction module. This component will be the base on which the other components such as the balancing module, and the persuasion module will rely on. After, the work presents the efficient bike trip module which main function is to evaluate the possible origin/destination stations and generate arguments or incentives in favor of these trips.

1.1 Motivation

Bicycle-sharing systems are a rising trend all over the world. Despite its existence for several decades they experienced an immense growth in the last years thanks to the adoption of information technologies which made possible a functional system for a large infrastructures and a large number of users. We can find this kind of vehicle-sharing system in over one thousand cities worldwide, spread in more than fifty countries with an estimated amount of 800.000 bicycles and 37.500 stations, being China and Paris the largest systems and Spain the country with most systems implemented [3, 4].

As mentioned, this kind of transportation systems offer some important advantages from individual motorized vehicles. To users, they provide a healthier and more affordable transportation method. To traffic, they help to control and optimize the number of vehicles, and reduce traffic jams. And to cities in which they are implemented, reducing the expending in big infrastructures (expand roads, parks for cars and large vehicles, etc) and leaving more space in te city to be reused for other purposes. In consequence shared vehicles, especially bikes, help to lower pollution levels. However, as previously stated, these systems also generate certain problems such as poor vehicle balance in different areas making it difficult to park or borrow bikes in certain circumstances. From the user perspective, this can generate dissatisfaction with the service resulting in a loss of subscribers which may lead to returning to the use of non-shared vehicles generating again the corrected disadvantages. On the other hand, from the service providers perspective, it can generate additional cost. Bikes must be rebalanced manually, involving additional workforce and fuel expense which also leads to increasing traffic and pollution levels if re-balancing is not done properly.

For all these reasons it is important to solve the mentioned problems in an efficient and inexpensive way, trying not to resort to extra work or big investments in order to offer an appropriate and sustainable service.

1.2 Objectives

The aim of this work is to design a multi-agent system capable of provide the self-balancing ability for a shared-bike service. To achieve this goal we set the following objectives:

- Study the state-of-the-art in order to know the current contributions and to acquire more knowledge about the presented problem.
- Collect data from our case study (Valencia bike sharing system).
- Analyze the collected data to obtain a better understanding of the service. Extract the valuable information and prepare it to the system use.
- Design a bikes and parking availability prediction module. This module must be able to predict the number of available bikes (or the number of free parking slots) for a station in a time frame.
- Implement the prediction module and test its correct functioning in order to ensure a minimum quality of the forecasting. Module results will be compared with two benchmarks: naive forecasting and average.
- Design an efficient bike trip module. This module will analyze origin/destination station pairs and select them generating arguments or incentives that will be presented to the users in order to try to slightly modify their behavior in favor of the correct operation of the system.

1.3 Work's structure

The work is structured as follows: in Chapter 2 we will talk about the related work. In chapter 3 the proposed architecture to solve the problem is discussed. Chapter 4 describes our case of study, focused in the Valencia's bike sharing system (Spain). In chapter 5 the prediction module is presented and discussed. Then the experiments and results for the prediction module are shown and discussed. In Chapter 6 the efficient bike trip module design is presented. Finally, in Section 7 we present the conclusions of this paper and in Chapter 8 discuss future work.

CHAPTER 2
Related work

Intelligent transportation systems have captured the attention of the AI research community in the last few years. The scale and complexity of the problems faced by the transportation system preclude simple and classic solutions from achieving the desired outcomes, hence the necessity to adopt AI approaches. As a consequence, multiple areas in AI have proposed solutions to different transportation problems. As example we can find swarm based algorithms for traffic control and travel planning [20, 37, 36, 23], driver behavior analysis for intelligent transport systems [9], vehicle managing using artificial intelligence [32], route planning in transport networks [6], etc.

In the specific field of multi-agent systems, the number of papers devoted to applications in traffic and transportation engineering has grown enormously. Bazzan and Klugl [7] present a literature review related to the areas of agent-based traffic modeling and simulation, and agent-based traffic control and management applied to different problems.

Focusing on the domain of bike-sharing systems we can find a description and discussion of this type of systems in [13]. In this work the author classifies the systems in four types, analyzes their particularities and discusses its potential and future possibilities.

Most of the problems that bike sharing systems arise are well studied. Finding optimal locations for bike sharing systems in order to be used to its potential needs to consider a huge amounts of circumstances: population density, employment density, proximity to universities, retail and commercial activity, access to bicycle infrastructure, proximity to tourist and recreation attractions, and proximity to other available transit options. This problem is discussed in [21] for the case of Richmond (Virginia), in [12] for the case of Milan, etc. Knowing some factors such as transport network status may be decisive in order to better understand certain vehicle sharing system functioning and establish patterns. Rebollo et al proposed an analysis module for transport network at [29] prepared to use information from several sources such as vehicle sensors, cities infrastructures, etc. and combine it with information that can be obtained from social networks, phone call registers and open portals. This module takes into account information of all the different transport media creating a multi-layer network that can model how they can affect each other.

We find two main problems that have been tackled by the AI research community: (i) predicting the bike/parking availability, and (ii) optimizing the transportation routing used to balance the bikes/parking positions across stations.

The first problem consists of predicting the number of available bikes and parking positions in the future. The rationale behind this is that, in order to improve the service given by bike sharing systems, users should be able to borrow and leave a bike when needed. Otherwise, users may become dissatisfied with the service and decide to use other transportation methods. The data mining community has made several efforts in this regard. Yoon et al. [39] propose prediction algorithms to predict the number of available bikes at origin and destination stations in Dublin's bike sharing system. The authors propose a modification of the ARIMA model to include information from neighboring stations along with the classic temporal information. The authors trained and tested their approach using approximately one month of data per process. While the authors employ important variables such as the available data in neighbor stations, they do not include other well-known factors that impact the usage of bikes such as weather data, nor they account for longer term seasonality such as seasons (e.g., summer, winter, etc.). Li et al. [24] proposed a multilayer data mining approach to predict bike traffic (i.e., bikes in transit) in New York City and Washington D.C.. In their approach, bike sharing stations are clustered together according to both geo-location and transit matrices. The advantage of employing clustering techniques to group stations is that predicting the traffic demand on the overall system and clusters is more robust and accurate than on individual stations. The multilayer approach first predicts overall traffic by using gradient boosted regression trees, and then distributes the overall traffic across cluster according to similarity between past and current data. Finally, the traffic between clusters is also predicted according to historic data. Differently to us, they possess bike trips information (i.e, bike id, origin, destination) whereas our dataset is solely composed by the current state of stations. That preclude us from employing the same approach.

In [28], Raviv and Kolka study the inventory management of a bike-sharing station analytically, presenting an user dissatisfaction function. In [11], Fernández experiments with bike availability forecasting for the bikes sharing system in Valencia (Spain), our case study. Fernandez-Vazquez tries lineal regression and compares it to k-nearest neighbors algorithm. In this work the forecast is hourly aggregated and only shows results for one day, what we think is not enough to test all the variables involved in the system behavior such as stationarity, weekday, etc. In [22], present bike/parking availability data combined with weather data for 27 cities across the globe (including Valencia, Spain). The authors analyze the correlation between weather data and bike demand, and find correlations between temperature, wind velocity, and precipitation with the demand of bikes in different cities. The paper does not propose any prediction mechanism per se, but it finds interesting effects on the bike demand, such as the effect of weather conditions, that are present in Valencia and other cities of the world.

The other main strand in research revolves around the idea of optimizing the routes and trips of the vehicles that balance bikes across the different stations. Given the nature of the problem, many researches have proposed the used of search & optimization techniques for this purpose. For instance, O'Mahony and Shmoys employ integer programming to balance bikes overnight, preparing them for rush time, and mid-rush balancing in New York City. In [17] is described the importance of bike repositioning at the stations in bike sharing systems. Having stations with no available bikes or with no free parking slots can lead to user dissatisfacition and a loss of service suscribers. To adress such scenarios Gosh propose an online and robust approach to better match the demand that usually does not have a well defined pattern. In the proposal an algorithm that use historical data modeling scenarios and online data is used. The algorithm consists in a two player game confrontation. In each iteration one of the players propose a situation that maximizes the loss and the other generates solutions minimizing it. When both players objectives converge an optimal solution is found. This metodology relies in the use of carrier vehicles from the service provider. In [16] Gosh extends the redistribution approach including a route planning for the carrier vehicles in order to reduce costs, environment issues and congestion while redeployment bikes. Lowalekar [25] provides a Lagrangian decomposition approach (that decouples the global problem into routing and repositioning slaves and employs using a new dynamic programming approach to efficiently solve routing slave) and a greedy on-line anticipatory heuristic to solve large scale problems effectively and efficiently.

As another example, Schuijbroek et al. [34] present a system that both predicts station demand and balances bikes attending to expected demand and desirable service levels. For prediction, authors rely on queues models and Markov chains using arrival and departure data from Boston and Washington. Due to the intractable nature of the routing problem when the number of stations is large, the authors cluster stations together so that service levels are guaranteed using only within-cluster routing. Then, mixed integer programming is used over the clustered problem. In order to partially-solve the carrier vehicles problems in [15] Gosh propose an approach for incentive users to fulfill redeployment tasks is presented. The rationale of the proposal is to give to the users that can be interested payment/trip based incentives. This way the author pretends to make less necessary the use of carrier vehicles. In [35] the authors present a complete architecture following this idea. While the idea can reduce the costs that service providers have to face in the bike redeployment task and make them more environmental friendly, this proposal does not take advantage of the very displacement that users do while using the service.

All of these system relies on prediction and optimization from the system designer perspective. The main difference between these approaches and our proposed architecture is that we rely on both a combination of optimization from the system designer, but also from the user perspective. In order to incorporate the users' actions into the optimization loop, we plan to use technologies such as incentives, persuasion [10], and negotiation [31].

AI research community also has interest in monitoring and controlling fleets. In [8] an agent support framework for fleet management is defined. This framework is specially suitable when working with open fleets. As defined in the article, bike sharing services belong to this category. Because of this reason we use this framework as a base, using the tools that this framework provides, for our multi-agent system.

CHAPTER 3
A general MAS proposal for bike sharing

As mentioned in Chapter 1, our aim is that of providing a MAS system for efficiently managing resources (i.e, bikes, stations, transportation trucks, etc.) in bike-sharing systems. The problem of optimizing bike-sharing systems is that of making sure that bikes are available in stations when users decide to start their trips, and parking positions are available when users reach their destinations. Due to the nature of cities and their lifestyle, bikes and parking positions become unequally distributed across stations. In order to cope with that situation, the service provider needs to redistribute bikes making use of transportation trucks. However, late distribution of bikes may end up in user dissatisfaction. Therefore, the real challenge for service providers is predicting future demand to redistribute bikes accordingly.

Balance operations carried out by the service provider will always be an integral part of bike sharing system, specially for preparing for rush hour. However, in some scenarios we may be able to employ users as balancing agents if individuals are persuaded to slightly deviate[1] from their planned destination/origin. The reasons by which these users may be persuaded vary and include reasons such as the fact that their destination station may be full at arrival, the adoption of healthier habits, or the inclusion of small rewards (e.g., extra rental minutes, badges, lotteries, etc.). Small deviations can act in benefit of the system by carrying out pre/after rush hour balancing, and acting as real time balance for unplanned demands.

In order to tackle this scenario, we propose a multi-agent based architecture. The proposed system will run on top of *SURF* [8], an agent support framework for open fleet management. The work we are presenting in this work is part of a broader research project, in which the main goal is to provide a set of tools and applications that foster the efficient and sustainable management of urban fleets. One of such applications is the one presented in [18] for last mile delivery in urban areas.

[1] We would never expect drastic deviations

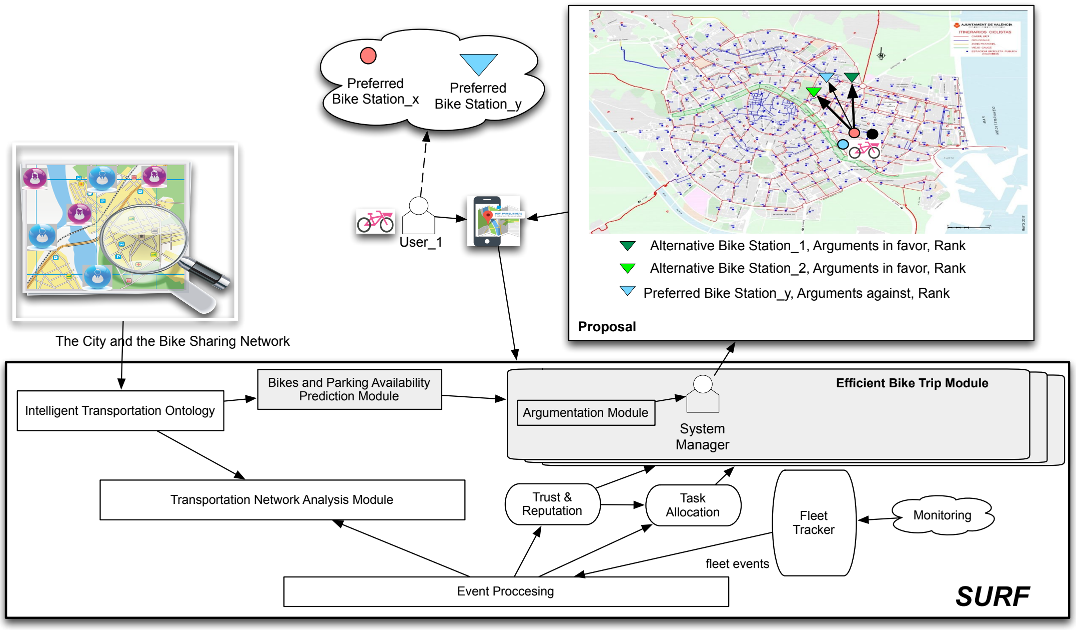


Figure 3.1: General view of the proposed MAS architecture

3.1 SURF framework

SURF is an agent support framework for fleet management. Classifying the fleets within the three fleet types proposed and discussed on [8] (static fleets, dynamic fleets and open fleets), *SURF* is designed and more suitable to use for the management of open fleets. Open fleets meets the following characteristics:

- Dynamic service demand: Service tasks may appear dynamically at any time and at any location.
- Dynamic number of vehicles: The fleet is composed by a dynamic number of vehicles. Even if the service has a prior fixed number of bikes it can continue offering service regardless the number of vehicles in any given moment.
- Autonomy/limited control: The fleet operator capability to control the behavior of the fleet may be limited since the availability and usage of a particular fleet's vehicle may depend also on the user or the owner.
- Size: Fleet functioning is not delimited by the scale being capable to potentially operate in a larger scale (maybe unlimited) than static od dynamic fleets.

We can take advantage of this peculiarity since bike sharing services can be considered a fleet of the open type. *SURF* framework is composed by a set of services and utilities to be used for fleet management. Can be divided in three different layers that groups functionality. First there is the fleet operator layer, in charge of fleet control and monitoring. Second the fleet coordination layer which function is to ensure the coordination. Lastly, the agent layer which connects with the agents that provides their information to the system.

SURF was designed to support general urban transportation fleets, and it provides modules for most general and shared functionalities. As a result, part of the proposed architecture is supported by these general modules. However, we need to include some extra modules to support some of the particular functionalities of this bike sharing system. Figure 3.1 shows the general view of the proposed architecture, the gray components being the modules specially tailored for the application of bike sharing, and the other modules being part of the services and utilities provided by *SURF* for open fleet management. The two main components that distinguish our approach to bike sharing are: **The Efficient Bike Trip Module** and the **Bikes and Parking Availability Prediction Module**.

The modules already provided by SURF framework that will be used in our proposal are the following: fleet tracker module, event processing module, trust and reputation module, transportation network analysis module and the intelligent transportation ontology.

The *fleet tracker module* is in charge of controlling the status and position of the vehicles. This data can be gathered from diverse sources: GPS location from user/vehicles, vehicle's status sensors and stations status information. Using this information the module will report to the system changes in the fleet giving to it more knowledge about the bike sharing system and the environment in order to make better decisions.

The *event processing module* receives the events generated by the fleet tracker module and the new task events generated by an user service demand. Its function is to analyze these incoming events in order to determine if a recalculation of tasks assignment is needed.

The *task allocation module* recalculates the assignment of all the pending tasks ensuring that this new generated allocation is the optimal global based on a set of assignment criteria.

The *trust and reputation module* has the mission to convert into a model the behavioral pattern of the agents. The models and information generated by this module is used in the task allocation module giving it more information that helps to determine the actions to be carried out. Also, the output of this module can be used to determine in which way the action taken can influence the agents. The clearest case in our approach is how the argumentation module can use this information in order to persuade or to offer an incentive to an user depending on his/her expected behavior.

The *transportation network analysis module* is responsible the analysis of the transport network. The module uses information from each transport media infrastructure and usage. In addition the module takes advantage of the data obtained from social networks, phone call registers, open data portals, etc. Building, with all this information, a multilayer network capable of consider the effects produced form one media to the others.In our approach is used to calculate the route and duration of the trips. This way the module will be able to estimate the arrival time to both, preferred origin and preferred destination stations. To achieve this, the module will use all the available information provided to the system by the user (GPS location, preferences, etc), city and the bike sharing network (routes, traffic status, traffic lights, weather, roads conditions, etc). The information will be retrieved from the intelligent transport ontology.

3.2 SURF extension

To these modules provided by the *SURF* framework we need to add two new modules to support our approach to the system. The bikes and parking availability module and the efficient trip bike module are crucial in order to include in the system a re-balancing mechanism based in the user behavior.

The **Bikes and Parking Availability Module** is in charge of making forecasts of the available free bikes or free parking slots in any bike station of the system. This module generated data will provide the MAS a very valuable extra information about the probable future status of some agents/elements of the bike sharing system. Taking advantage of the prediction ability of the module the system will be able to generate a set of alternative routes. These alternative routes are computed keeping in mind the departure and arrival stations if the original ones selected by the user are considered, following some criteria, not optimal for the system general functioning.

The **Efficient Bike Trip Module** aim is to collect the set of alternative routes generated by the bikes and parking availability module and analyze them. The module will score every alternative in order to select the propitious for the system, order them from the best to the worst and generate argumentation or incentives in favor of each alternative.

Both the **Bikes and Parking Availability Prediction Module** and the **Efficient Bike Trip Module** support how users' trips are managed. In order to understand the logic behind the module, let us focus on an example:

1. $User_1$ agent wants to ride from the station $PreferredBikeStationx$ to the station $PreferredBikeStationy$. The user employs a mobile app to query the availability of bikes at the origin station, and the availability of slots in the destination station.

2. The request is received by the System Manager agent, and then it is analyzed to find out the availability by the time $User_1$ agent may arrive to both preferred origin and destination stations. The expected times are calculated taking into consideration the current GPS location of $User_1$

agent, the possible route that leads to the origin station, the possible route that leads to the destination station, and all the information from the Intelligent Transportation Ontology from SURF concerning traffic, traffic lights, weather, and so forth.

3. With this time frame the System Manager agent requests to the **Bikes and Parking Availability Prediction Module** an estimation for the number of free bikes at $PreferredBikeStation_x$ by the expected departure time. At the same time, the System Manager agent also requests an estimation for the number of free parking slots at $PreferredBikeStation_y$ by the expected arrival time.

4. The prediction module also computes whether or not the station $PreferredBikeStation_x$ or the station $PreferredBikeStation_y$ are likely to suffer from bikes/slots shortage in the short/medium term. In that case, the prediction module retrieves a set of available nearby stations to station $PreferredBikeStation_x$ and a set of available nearby stations to station $PreferredBikeStation_y$. If they are not likely to suffer from bikes/slots shortage in the short term, then they are also suggested to the System Manager agent.

5. The System Manager agent collects the suggestions from the **Bikes and Parking Availability Prediction Module** and sends those suggestions to the **Efficient Bike Trip Module**. Within this module, the alternatives for both origin and destination are analyzed. The module will select pairs of origin and destination stations, along with arguments or incentives in favor of the slight trip change.

6. The System Manager agent receives the offers from the **Efficient Bike Trip Module** and presents them to the user, who finally selects the one that he/she considers more appealing.

In this work, we focus on the **Bikes and Parking Availability Prediction Module** and the **Efficient Bike Trip Module**. More specifically, we focus on developing and finding prediction models for bike availability at different stations in Valencia's bike sharing system, our test scenario. This module is the cornerstone to the application of Bike Sharing in Urban Areas, as its outputs are needed to compute the availability of preferred and nearby stations. These stations are later used as building blocks for building arguments in the **Efficient Bike Trip Module**. The following sections describe in detail the data mining modeling carried out and the performance of the proposed prediction approach with a use case from the bike sharing system of Valencia. Also, our initial approach for the **Efficient Bike Trip Module** is described and discussed.

CHAPTER 4
Case study: Valencia's bike sharing system

The aforementioned architecture is abstract and general, making it applicable to a wide range of urban systems and cities. Nevertheless, as part of the verification of the architecture, we decided to test the proposed architecture in some realistic scenarios. The focus of this paper describes the on-going work in one of those realistic scenarios. More specifically, we focus on the application of the architecture to Valencia's bike sharing system. The reasons to focus on this domain are varied: access to domain expertise, possibility of linking the bike sharing system with other urban transportation methods, access to data, and the scale of the proposed system.

Valencia's population is close to 800,000 inhabitants, and it exceeds the 1.5 million inhabitants when considering its metropolitan area [1]. This makes Valencia a large/medium-sized city, which makes it appropriate for the verification of our architecture. On top of that, its flat landscape and availability of dedicated bike lanes foster the use of bikes as an urban transportation method. Valencia's bike sharing system consists of 276 bike stations whose capacity varies between 14 and 50 slots, with an average of 20 slots per station. Therefore, there are 5,500 parking slots for a total of 2,750 bikes available to users. The operations of the bike sharing system started in 2011, rapidly gaining around 100.000 subscribers in it's first two years. In the next two years, as seen in most of these bike sharing systems, the number of user subscriptions dropped and stabilized around 45,000 users [2]. Figure 4.1 shows the distribution of the bike sharing stations in the city.

We have collected open access data from all the stations in Valencia[1], containing information about the number of slots and bikes available at each station. This information is collected periodically with a frequency ranging from one to ten minutes[2]. In total, we have collected 617 days of activity starting from 26th September 2014 to 15th February 2017. This results in a total of

[1] http://gobiernoabierto.valencia.es/en/

[2] Sometimes technical issues and systems overload preclude from sampling at the same frequency

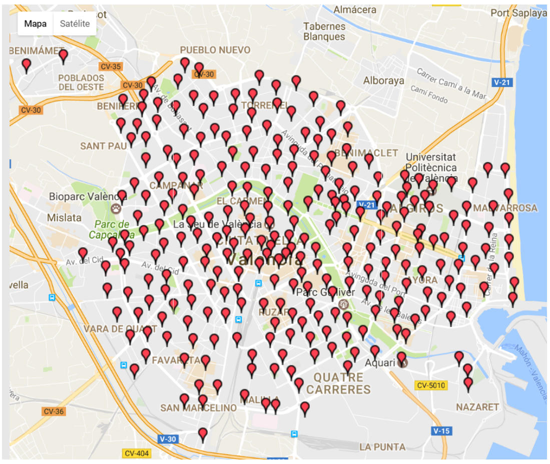


Figure 4.1: Bike sharing stations in Valencia, Spain

62,130,711 records containing information about the occupation of a station in a particular point in time.

As suggested by [22], weather conditions may influence the demand for bike sharing systems. As a consequence, we collected information about the weather conditions[3] in Valencia, including attributes such as temperature, rainfall and wind speed. The information is collected with a granularity of 30 minutes and then merged with the station data. In total, we have collected 1141 days of weather data starting from 1st January 2014 to 15th February 2017.

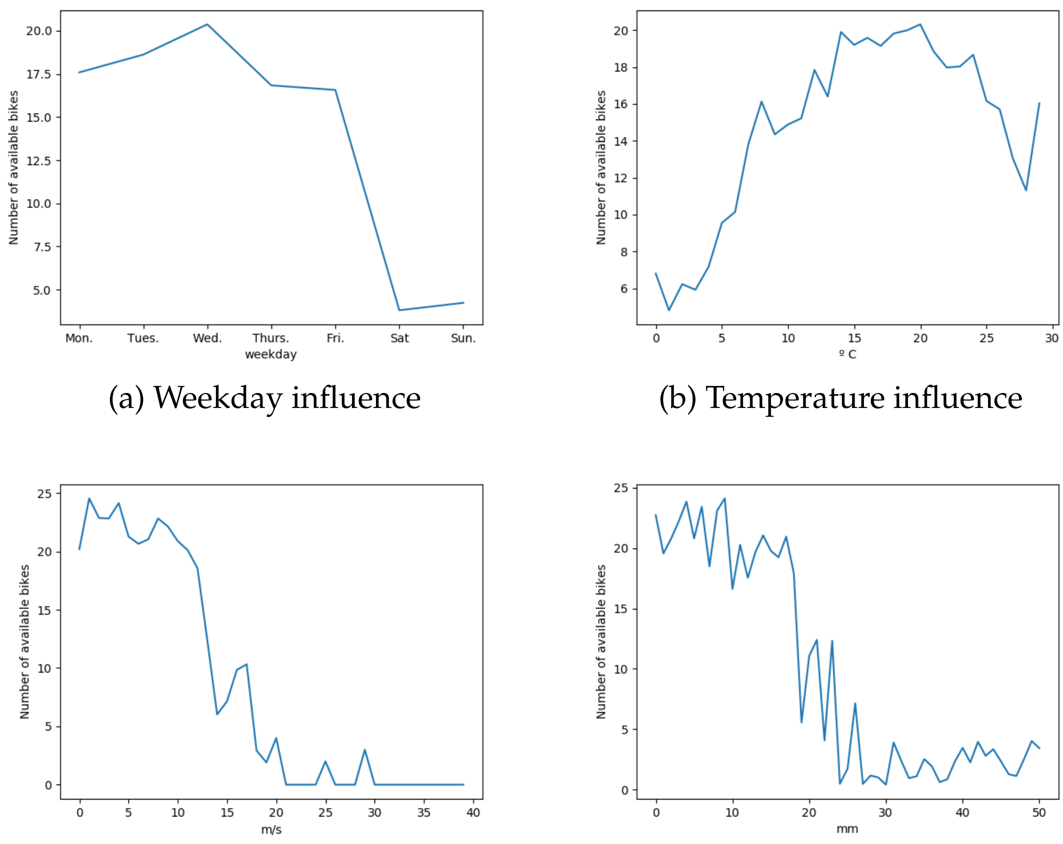
We merged together both data sources, resulting in a single dataset whose samples contained information about the status of the station and weather conditions at a certain timestamp. With this dataset, we endeavored to analyze what variables could help us with the task of predicting bike usage in our case study.

4.1 Data analysis

Firstly, we attempted to analyze whether or not the day of the week could influence bike demand. Our initial hypothesis was that the day of the week would influence how people move around the city. During the week, stations in popular work areas are most likely to receive incoming and outgoing traffic than during the weekend. Similarly, leisure areas are more likely to receive traffic during the weekend. With that goal in mind, we plotted the average number of available bikes for each day of the week. In Figure 4.2 (a), it can be appreciated that our reasoning was correct. The figure shows the average number of available bikes for *UPV Informática*, one of the most transited bike stations due to its proximity to one of the largest universities in the city. It is shown that, during the weekend, barely no bikes are available at the sta-

[3] https://www.wunderground.com/

tion, while the rest of the week the station acts like a sink. This behavior was aligned with common sense, as universities tend to be more active on the weekdays. Although not shown in the graph, we could observe this and similar patterns in other stations throughout the city.



(a) Weekday influence

(b) Temperature influence

(c) Wind speed influence

(d) Rainfall influence

Figure 4.2: Influence of environmental variables

Then we proceeded to analyze the influence of temperature on bike demand. Our initial hypothesis was that colder and extremely warm days are less propitious for riding bikes, specially in days when environmental conditions are harsher. Those days, individuals are most likely to refrain from using the bike sharing system and use other transportation methods that are more sheltered from the outside conditions. With that in mind, we plotted the average number of available bikes at *UPV Informática* during the daytime. Figure 4.2 (b) shows our initial hypothesis. Before analyzing the graph, one must consider that this station usually acts as a sink during the daytime. Therefore, reduced demand is translated into less bikes arriving to the station. This is exactly what is shown in the figure. In colder days, demand tends to be minimum, and it gradually increases as temperature becomes more comfortable. There is again another drop in the demand when days become hotter. Despite not being shown in the graph, we could observe this behavior in other stations throughout the city.

Our rationale for the wind speed was similar. Stronger winds make it difficult to handle bikes, with even the risk of falling off in case of a very strong gust of wind. Hence, users may be more hesitant to use the bike sharing system in those particular days. We plotted a similar graphic to describe the relation between the bike demand and the wind speed. Figure 4.2 (c) shows the relationship between the average number of bikes at *UPV Informática* during the daytime for different ranges of wind speed. As we expected, bike demand in the station is reduced as the wind speed increases, supporting our initial guess. Again, we found a similar pattern in other stations.

Following our thoughts regarding the effect of wind speed on bike demand, we made a similar conjecture with regards to rainfall precipitation. When rain is absent, users should employ the system as usual. However, as rain becomes more prominent, demand should decrease since users will feel less comfortable riding a bike. In extreme conditions, rain may make the ground slippery, thus making bike riding a dangerous activity. In Figure 4.2 (d) the average number of available bikes at *UPV Informática* during daytime is shown for different levels of precipitation. Our rationale was again supported by data. For no rain or light rain the station's demand is unaffected, but less bikes tend to arrive (thus, reducing demand) when the rain becomes heavier.

We analyzed season influence on bike demand. The hypothesis was that seasons with more extreme weather conditions should have a notable minor average use than the seasons with a more cozy conditions. Besides, the season with more significant dates such as holiday (specially longest ones as summer holiday, Christmas...) should have a remarkable impact on the demand. In Figure 4.3 can be observed that spring and autumn have a higher demand than winter and summer probably due, as said before, to the less favorable weather conditions and to being the two seasons with more number of non-working days.

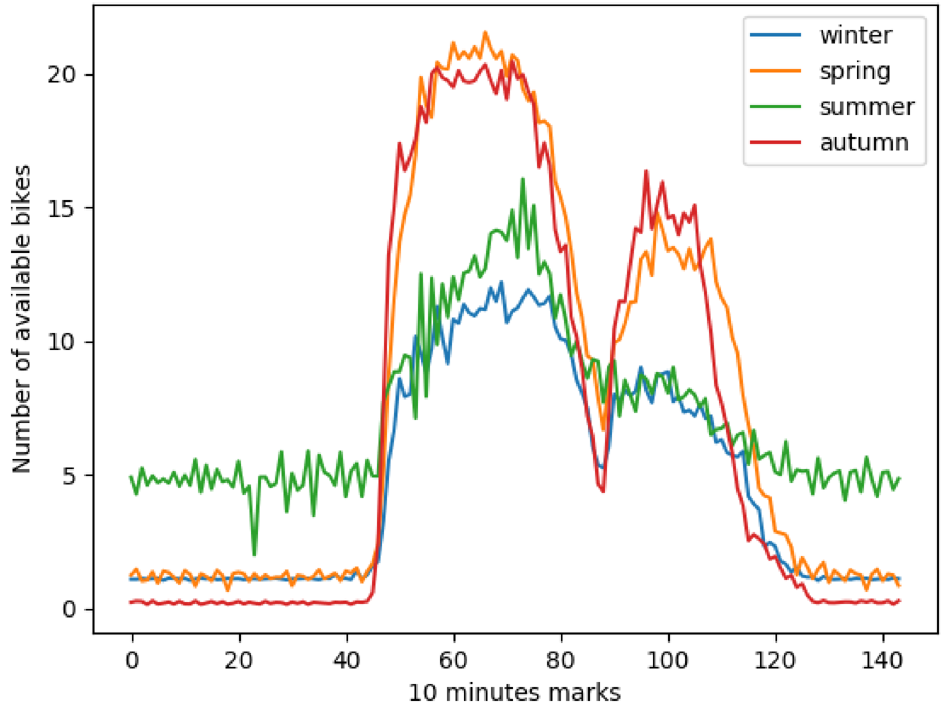


Figure 4.3: Influence of season

Lastly, we had the hypothesis that it would be very likely that social events influence on bike demand. Out thoughts were that the fact that social events attract many people was enough to increase bike-sharing usage. Added to this, the elevated number of people attending makes it very difficult to arrive because of the jams produced by the traffic increase or park due to the large increment of people in the same place. This conditions make public shared bikes a perfect alternative to attend some kind of events. To prove our guess, we analyzed the surrounding stations of *Mestalla's* football stadium, more concretely the seven closest stations. In the plots 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 and 4.10 is represented the number of available bikes during a match day and during a non-match day [4]. The vertical lines represent the time of start and finish of the match. Is clear that moments before a match a peak in the number of bikes starts to form. The peak starts to vanishing when the match arrives to its end. In the day with no football match this peak is non-existent, confirming our hypotheses.

[4]The non-match (24/4/2016) day was chosen to be the same weekday than the match day but a week before (1/5/2016) in order to avoid variable interference in the data

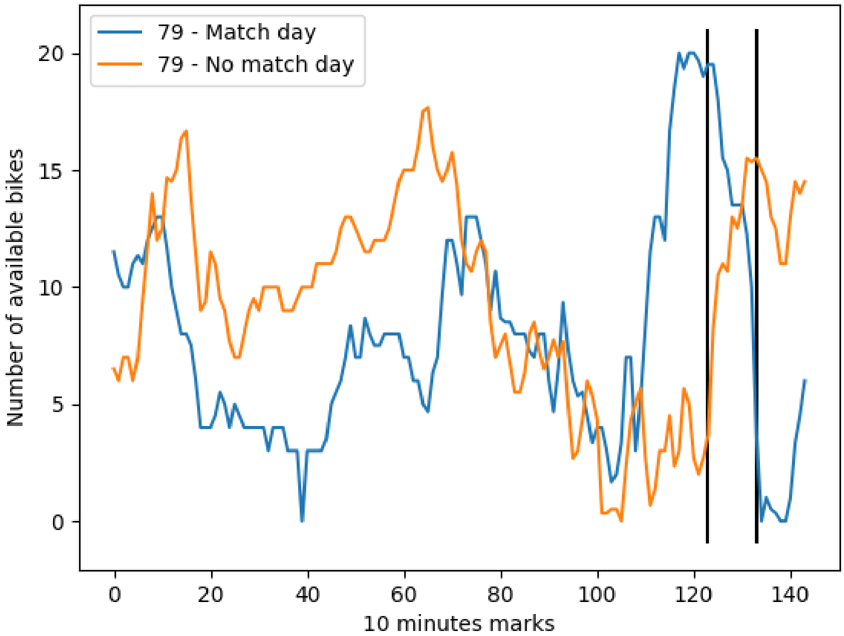


Figure 4.4: Football match influence in Aragón - Ernesto Ferrer station

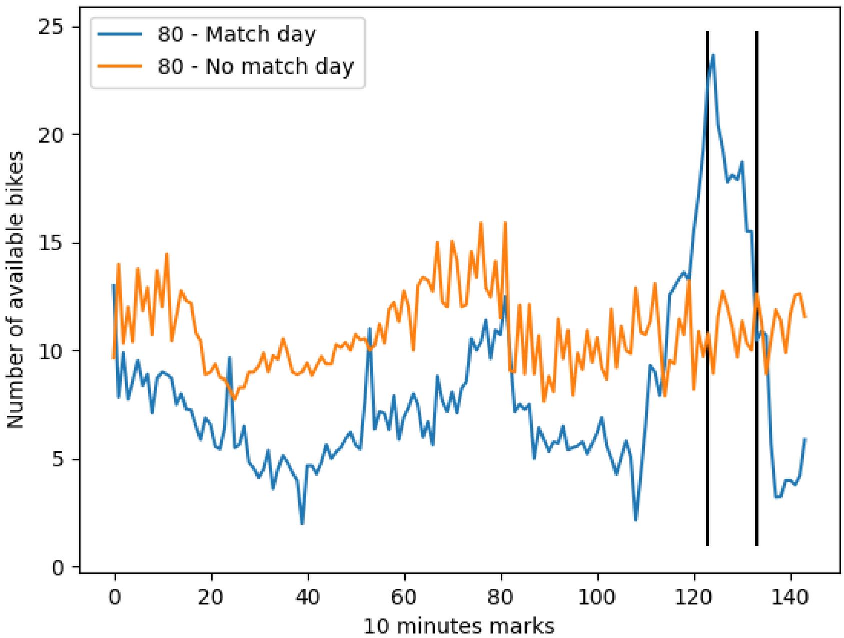


Figure 4.5: Football match influence in Amadeo de Saboya station

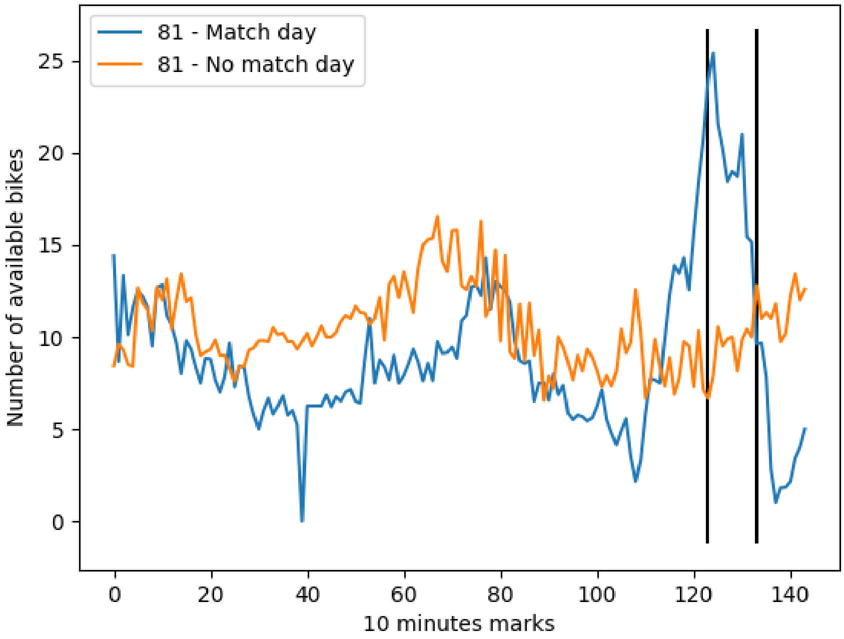


Figure 4.6: Football match influence in Micer Mascó - Rodriguez Fornos station

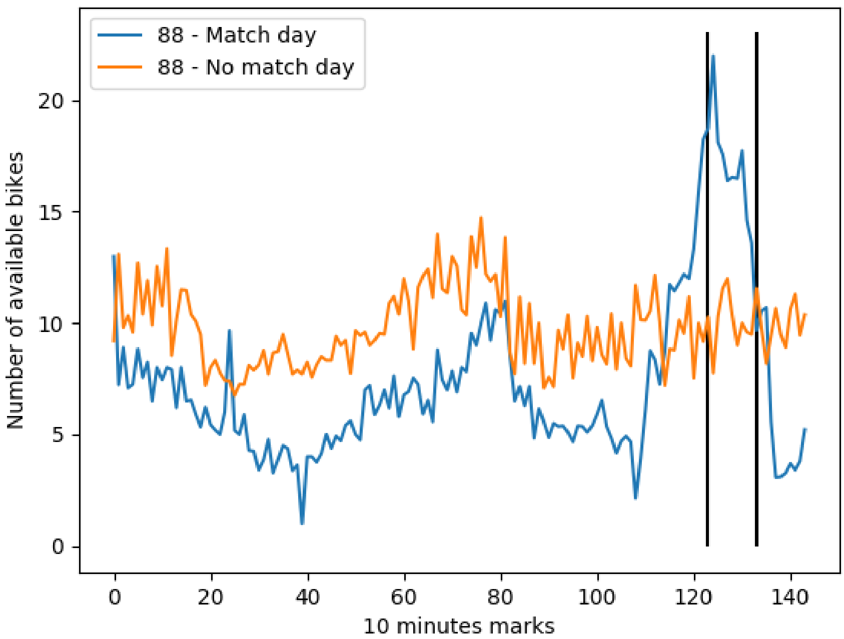


Figure 4.7: Football match influence in Blasco Ibañez, 28 (F. Geografía e Historia) station

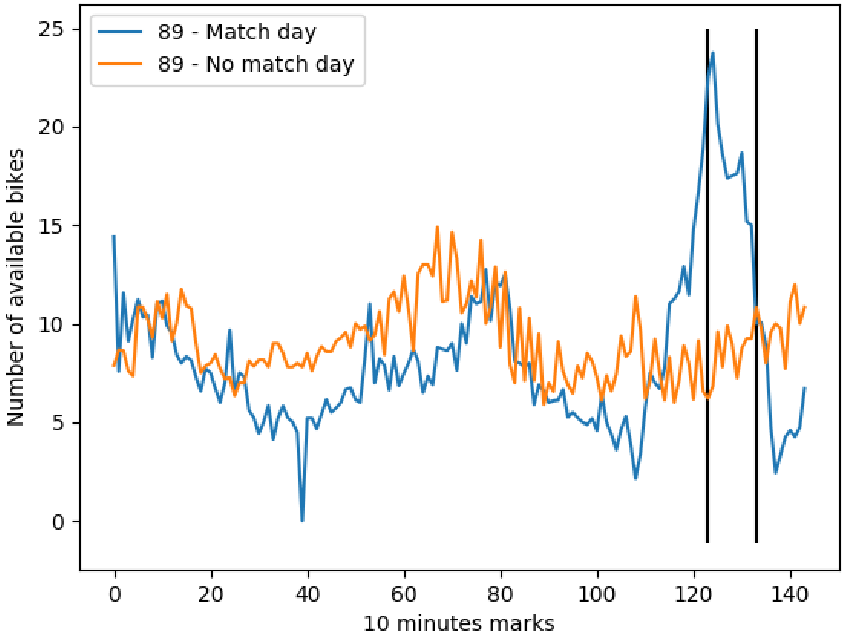


Figure 4.8: Football match influence in Blasco Ibañez, 23 (F. Filosofía y Psicología) station

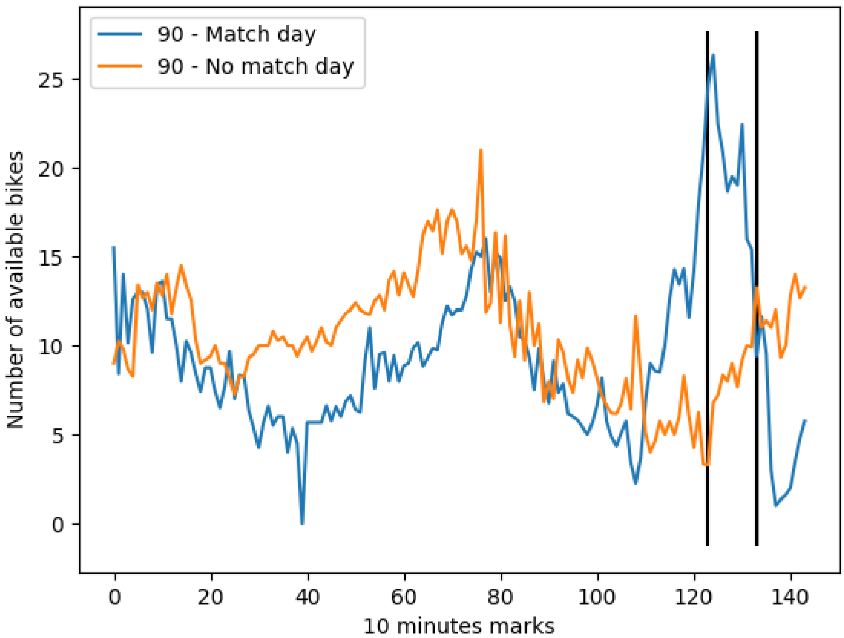


Figure 4.9: Football match influence in Blasco Ibañez, 32 (F. Filología) station

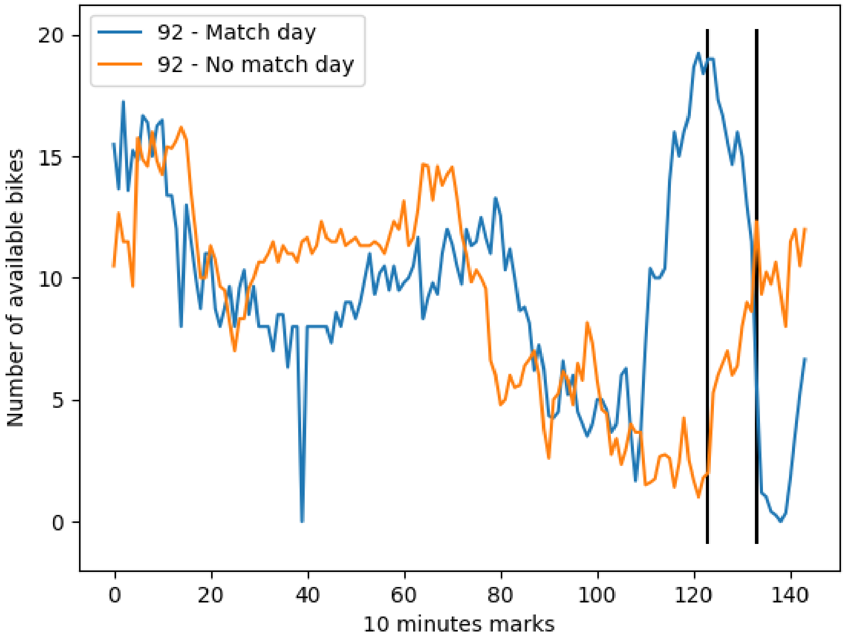


Figure 4.10: Football match influence in Blasco Ibañez - Aragón station

All of these insights were taken into consideration when deciding what variables should be part of the final dataset used for training our prediction module. More specifically, the records in the resulting dataset consisted of a station id, a timestamp decomposed into year, month , day, hour, minute, second, and weekday of the measurement, temperature, rainfall precipitation, wind speed, and the number of free parking slots and bikes available in the station. We discarded as variables the season and the events. The season, although being influential on the case, can be inferred by the date. In the case of the events information the decision of excluding them was caused because the very small amount of data that we were able to acquire. This data is also limited for a very small number of stations.

The next Chapter describes how we trained different machine learning models for obtaining accurate predictions in different stations across the city.

CHAPTER 5
Bikes and parking availability prediction module

The bikes and parking availability module has the function of forecasting the number of available bikes (or the number of the free parking slots) for every station in a time frame. This module is crucial for the system and to obtain an improvement in public bicycle rental services. Since the module main function is to predict the future status of the stations and, if needed, a set of alternative routes it will contribute to the system with a very valuable data. This data will be very helpful to other modules, specially the efficient trip module. It will allow to the mentioned module to compute which alternative routes are better for the system having in mind the user guided balancing of the bike shared service, score them and generate arguments in its favor.

Trying to predict the state of the stations, either the number of available bicycles or the number of free parking slots, we are facing a regression problem. Regression techniques attempt to approximate an unknown function that would represent the "truth" of the situation to be studied. Since the real function is not knowable, this approximation is made trying to model the relationship between a set of explanatory variables with the dependent variable. In our case study, the explanatory variables are the selected in the analysis carried out in Chapter 4 (year, month, day, hour, minute, second, weekday, temperature, rainfall precipitation and wind speed) and the dependent variable is the number of free parking slots available at the bike station. In this work we have opted for the use of machine learning algorithms to design and implement the regression model for the module.

Once determined which variables may be useful for our prediction module, an experimental setup is prepared to test the accuracy of different machine learning models. First, we describe the general settings for the experiments. Then we describe and analyze how hyper-parameters are optimized for the machine learning algorithms. Finally, we analyze the performance of the best models on the test data.

5.1 Machine learning algorithms

5.1.1. Support Vector Regression (SVR)

Support Vector Machine is a supervised machine learning technique that can be used both for regression tasks and classification tasks. When used for classification problems it is commonly known as Support Vector Regression (SVR)[5].The SVR algorithm is a nonlinear generalization of the Generalized Portrait algorithm. The main idea of the SVR is to select the regressor hyperplane that better fits to the training dataset to achieve generalized performance. To accomplish this, SVR tries to minimize the general error bound using a loss function that is capable to quantify the error between the target value from the data and the approximation generated by the model. The loss function ignores the errors associated with the points that are between the regression function and a determined margin distance. The support vectors are the instances across the margin. In order for the model to solve non-linear problems it can be expanded using kernels.

5.1.2. Artificial Neural Network (ANN)

Artificial Neural Networks (ANN)[33] are computational models capable of learning relations between the input data and the output data in a supervised way. ANN's are composed by a collection of simple units, also called artificial neurons, connected together in a specific pattern. Every connection has an associated weight that determines how important is the output from the neuron at the beginning of the connection for the neuron at the end. Each of the artificial neurons have an associated transfer function that determines how that unit's value is updated (Figure 5.1). The transfer unit is composed of a net input function, and an activation function.

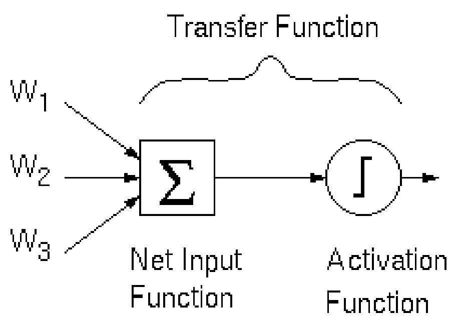


Figure 5.1: Neuron representation

The net input function determines how the net input is combined, usually adding all the inputs. The activation function determines how the neuron value is converted after its transferred to the next neuron. There are diverse activation functions that can be classified in two groups: linear functions and non-linear functions. Linear functions are commonly used in the input layer in order to feed the network with the data or in the output layer in regression tasks. Non-linear functions are used in the hidden layers in order to provide the network the capability of recognize non-linear behaviors or in the output layer in tasks such as classification using the softmax function. The unit receives one or more inputs from connections (directly from the data or from another unit). The connections weights are the parameters that the Neural Networks must learn.

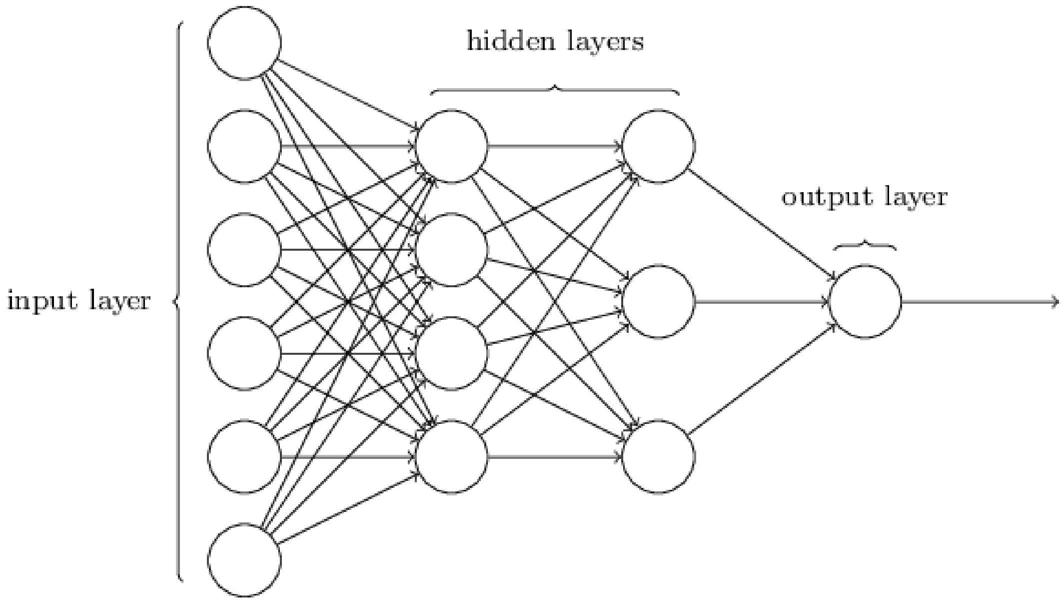


Figure 5.2: ANN topology example: Multi Layer Perceptron

There are different types of networks depending on their topology. Some of them are:

- Feedforward networks (FFN): in this type of network the information moves only forward. In other words, the connections between the artificial neurons exists only between the ones from a layer and the ones from the next layer without creating loops. Some examples are MLP or Multi Layer Perceptron (Figure 5.2), Autoencoders or Convolutional Networks.

- Recurrent networks (RNN): this kind of networks can propagate the information forward and also backwards creating loops. Examples of RNN's are Bi-directional Networks and Long Short-Term Memory

With the purpose of making ANN's learn, the network parameters must be learned automatically while feeding the network with data. In order to accomplish this ANN's use an algorithm named backpropagation. This method calculates the gradient of a loss function (which determines the performance of the model) with respect to the network parameters. Optimizers use the backpropagation algorithm to adjust the parameters optimizing this way the network performance. The optimization algorithm repeats a two phase cycle for every training sample presented to the network. First, in the forward phase, the input vector is propagated forward layer by layer until the information reaches the last layer generating the networks output. In the backpropagation phase the output is compared with the target output through the use of a loss function calculating the error for each neuron in the output layer. The error values are propagated backwards until the first layer. The back propagation algorithm uses this error to calculate the gradient of the loss function. This gradient is used by the optimization to update the parameters of the network trying to minimize the loss function. During the training process, the units of the hidden layers organize themselves becoming capable of identify different characteristics of the input space.

After the training the network will react to each presented input generating an output making use of the learned relation between the input data and the output data.

5.2 Experiment design

In order to achieve an appropriate performance in the proposed system, we need to obtain an accurate model that allows us to estimate the number of empty parking slots [1] for a station in a future time. Since we can use historical data to model users behavior, we are facing a regression problem. Machine learning algorithms have consistently proven to provide accurate regressions for a wide variety of domains [38, 26, 19]. Hence, we decided to approach this problem by considering two of the most successful machine learning regression algorithms: support vector regression, and artificial neural networks. Given the restriction of 30 minutes per trip that is established by the service provider, we decided to formulate a regression problem where, given the current state of a station and associated weather variables at that instant, we attempt to predict the number of bikes in the station in the next 30 minutes.

The dataset was divided into two parts: the training and the test dataset. The training dataset was exclusively used as a testbed for hyper-parameter tuning. Then, the performance of the best hyper-parameters was tested against the test set to assess the performance of the models in a realistic setting. Since the quality of the resulting model depends on the quantity and quality of the training dataset, 80% of the total data was employed as training, while the remaining 20% was left for testing purposes. This means that the final performance of the models was tested with approximately three months of data, amount that should be sufficient enough to ensure meaningful results. The test set consisted of the last three months of available data. As for measuring the performance of the model, we employed the average mean squared error (MSE).

[1] Equivalent to predicting the number of available bikes, as it can be obtained by subtracting the empty parking slots from the total number of slots

5.3 Hyper-parameter optimization

Hyper-parameters are the parameters of the model which value cannot be derived at the training process. Instead, this kind of parameter must be set before the commencement of the experimentation or the learning process. The choice of hyper-parameters is an important task due that depending on their values the machine learning algorithms can lead to undesired situations in the training step which can end into performance problems of the model.

Due to computational limitations, the fine tuning process was exclusively carried out in a single bike station: *UPV Informática*. The methodology employed for finding the best model hyper-parameters was a grid search over the space of possible values. Following, we describe the hyper-parameter space for each of the selected machine learning algorithms.

- **Support Vector Regression:** We decided to employ a radial basis function kernel in order to model non-linear relationships between the input variables. For the penalty error parameter (C), we tested values in the range of 10^{-5} and 10^{4} with increases in powers of base 10. γ was set between the range of 10^{-5} and 10^{4}, again increasing exponentially with base 10.

- **Artificial Neural Networks:** A 3 hidden-layer topology with ReLU activation functions was chosen for study, the last layer being the only one with a linear activation function. The neurons in each hidden layer varied exponentially from 8 to 2048 with a base of 2. On the other hand, the learning rate of the network also exponentially varied between 10^{-7} and 0.1, but this time the base being 10. Either stochastic gradient descent or RMSProp were employed to optimize the weights of the network.

The best artificial neural network was found to be a 3 hidden layers network with 64 nodes per hidden layer. The best learning rate was found to

neurons/lr	**1.0E-01**	**1.0E-02**	**1.0E-03**	**1.0E-04**	**1.0E-05**	**1.0E-06**	**1.0E-07**
8	79.26	11.97	5.15	5.25	5.45	74.21	705.14
16	74.99	9.52	5.01	5.01	5.42	72.13	675.10
32	76.71	8.70	4.20	5.70	4.96	42.26	499.00
64	78.26	14.05	**3.54**	3.81	4.99	6.01	75.64
128	76.43	76.13	5.69	4.69	4.95	5.38	74.18
256	79.36	13.63	5.28	3.71	4.80	5.04	72.64
512	77.31	11.54	5.17	4.33	5.29	5.04	28.38
1024	77.29	11.57	4.40	5.86	4.55	5.28	6.86
2048	75.74	11.54	4.32	4.23	3.99	6.55	5.31

Tabla 5.1: Results for the grid search carried out for RMSProp optimized neural networks. The best result is highlighted in bold font

be 0.01, with smaller learning rates providing almost constant predictions for any input (i.e., a sign that the network does not learn any pattern due to a slow convergence) and larger learning rates providing totally inaccurate predictions. The best network optimizer was found to be RMSProp.

With respect to the support vector regression, we found that the optimal value for the penalty error was 10^4, while the best value for γ was found to be 10^{-5}. In general, we found that artificial neural networks tended to produce more accurate predictions than support vector regressions. Detailed results for the grid search can be found in Tables 5.1 and 5.2.

5.4 Test results

Once we had two candidate models, we employed the test set to realistically assess the performance of the models in a deployed application. This time, instead of focusing on a sole station, we trained several stations coming from different city districts. This way, we can better study the accuracy of the models in a realistic setting. In addition to these models, we also introduced two benchmarks for comparability purposes. One of the benchmarks outputs the average number of bikes available in that station throughout history, whereas the other benchmark always outputs the current state of the station as a prediction. If trained correctly, both our models should outperform the benchmarks.

Table 5.3 shows the results obtained by the four prediction models using the test set. Those results that are statistically better according to a Mann-Whitney test with $\alpha = 0.05$ are highlighted with bold font. As it can be observed, the artificial neural network tends to outperform the rest of the models in almost every station tested. More specifically, the artificial neural network was the best choice for 16 out of the 19 stations (approx. 84% of the stations). The support vector regression was one of the best choices for only 8 of the stations, accounting for 42% of the scenarios. In none of the scenarios the benchmarks produced better predictions than the two machine learning models. This information is also represented in Table 5.4, where the relative improvement of the machine learning models versus the benchmarks are compared. As it is observed, the ANN model improves the predictions of the benchmark that predicts the current status by 16.51%, the benchmark that predicts the average bike availability by 63%, and the SVR model by 10.05%. Overall, it is the best performing model in these scenarios.

However, it should be noted that in some cases the prediction of the machine learning models and the benchmark that outputs the current state of the station are close. This suggests that some stations may require different hyperparameters to distance their outputs from benchmarks. As another sidenote, we observed that, in some stations, there is very little activity throughout the day. This means that a benchmark that outputs the current state of the stations, is also likely to produce accurate results many times. It will only pro-

C / γ	**1.00E-05**	**1.00E-04**	**1.00E-03**	**1.00E-02**	**1.00E-01**	**1.00E+00**	**1.00E+01**	**1.00E+02**	**1.00E+03**	**1.00E+04**
1.00E-05	148.40	148.40	148.39	148.34	148.40	148.40	148.40	115.47	115.47	115.47
1.00E-04	148.40	148.40	148.32	147.73	147.12	122.89	115.47	115.47	115.47	115.47
1.00E-03	148.40	148.33	147.55	140.80	135.27	122.29	115.47	115.47	115.47	115.47
1.00E-02	148.30	147.44	140.12	148.34	73.27	116.42	115.46	115.47	115.47	115.47
1.00E-01	147.35	139.15	77.87	27.75	20.45	84.46	115.42	115.47	115.47	115.47
1.00E+00	139.10	148.39	38.03	144.32	14.77	24.05	104.86	115.47	115.47	115.47
1.00E+01	86.27	22.52	279.17	79.11	14.38	18.58	75.89	102.00	115.47	115.47
1.00E+02	22.67	26.16	812.06	35.38	24.93	25.80	89.11	123.14	136.49	136.63
1.00E+03	32.55	21.27	77.22	87.01	42.44	31.40	89.11	123.14	136.49	136.63
1.00E+04	**5.10**	18.66	154.04	301.69	138.33	31.20	89.11	123.14	84.23	84.21

Tabla 5.2: Results for the grid search carried out for support vector regression. The best result is highlighted in bold font

duce inaccurate predictions in the few instants when a bike arrives or leaves the station.

Station	**ANN**	**SVR**	**Current**	**Average**
City hall - Cotanda	**6.01**	8.07	7.80	19.22
Colon station	**5.50**	**5.49**	6.60	13.86
Porta de la Mar	**5.80**	6.75	6.79	17.39
Plaza de los Fueros	8.08	**7.15**	8.50	18.54
Peris y Valero - Luis Santangel	**4.05**	5.01	4.67	9.33
Av. Puerto - Dr. Manuel Candela	**4.83**	**4.81**	5.44	7.35
Av. Puerto - Jose Aguilar	**2.89**	2.96	3.16	5.28
Molinell - Calderon de la Barca	**2.34**	3.16	2.59	10.63
Blasco Ibañez - Poeta Duran Tortajada	6.05	**5.72**	9.11	10.67
Blasco Ibañez, 121	**4.39**	**4.34**	5.42	8.29
UPV Caminos	**11.59**	15.03	13.74	31.46
UPV Informática	**3.82**	5.10	5.52	25.46
Benimaclet station	**5.49**	6.73	6.82	7.31
Turia station	**1.34**	1.43	1.41	21.76
Manuel Candela - Rodriguez de Cepeda	**4.53**	**4.60**	5.45	10.82
Reig Genovés - Ramón Contreras Mongrell	**1.70**	2.11	1.96	11.76
Hospital Nueva Fe	**3.16**	4.70	5.94	32.63
Giorgeta, 64	**1.29**	**1.28**	1.39	6.92
Veles e Vents	3.57	**3.53**	3.73	8.76

Tabla 5.3: Mean squared error for the four prediction models across different stations in the city

Table 5.3 shows the MSE for different stations distributed across the city. Despite the fact that lower MSEs indicate more accurate predictions, they are not very informative of the practical quality of the best predictive model per se. Therefore, we decided to plot the prediction of our ANN model against the target value for a given day. Figures 5.3 and 5.4 shows this comparison for the *City Hall - Cotanda station* and the *UPV - Informática station* respectively.

The figure suggests that, in both cases, the ANN model is capable of closely matching the real bike demand. This happens for most of the day, even matching some of the peaks in the bike demand. However, there are some sudden peaks that are not as closely matched as the rest. This suggests that some of the peaks may be accounted by other variables not necessarily included in our dataset. For instance, transportation trucks balance bikes across stations, in-

VS.	Current	Average	SVR
ANN	16.51%	63.05%	10.05
SVR	6.09%	59.47%	N/A

Tabla 5.4: Relative improvement of model (rows) versus benchmark (columns)

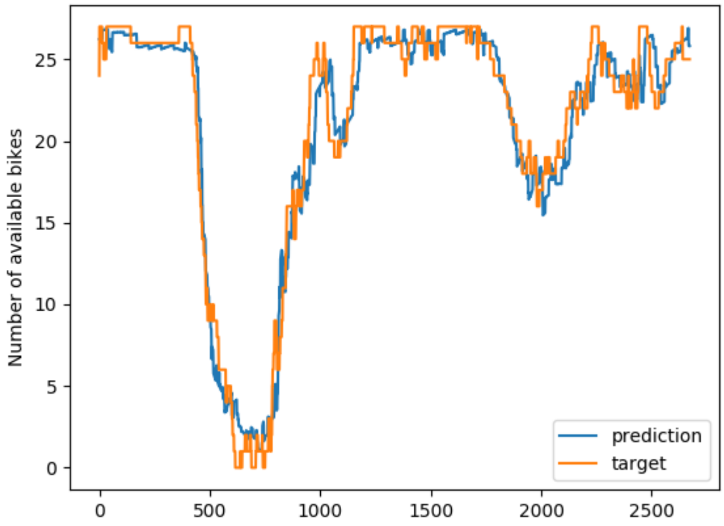


Figure 5.3: Prediction of the ANN model versus the target value for the City hall - Cotanda station

formation that is not included in open access datasets. This problem is also documented in other similar works [39, 22]. We are currently working on including more data sources to attempt to better predict some of these sudden outbursts of activity. For instance, we are including data about sport and music events, national, regional, and local holidays, and nearby transportation methods.

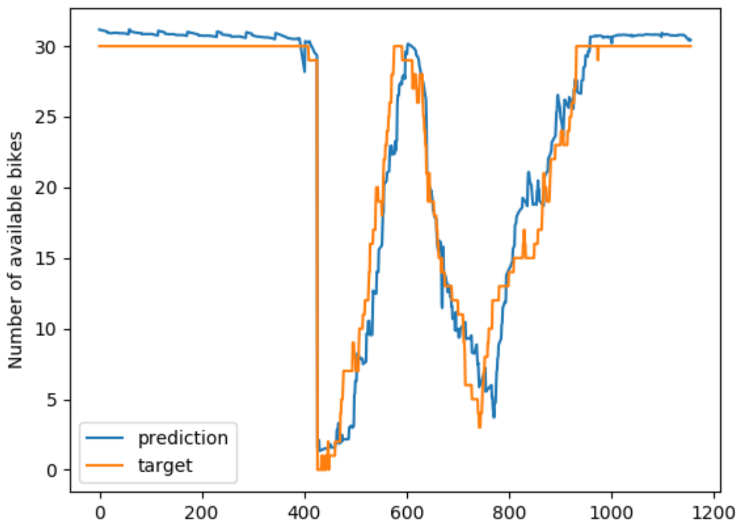


Figure 5.4: Prediction of the ANN model versus the target value for the UPV - Informática station

Overall, the experiments suggest that ANN is the best current candidate for predicting the bike demand in Valencia, Spain. The predictions also closely match the real demand. This is of crucial importance for our multi-agent system, since, as we described in Section 3, the prediction module is the base for the argumentation & negotiation module. We expect that this module will allow us to incentivize users and make them balancing agents that optimize the overall performance of the system.

CHAPTER 6
Efficient bike trip module

The efficient bike trip module is the second module designed as a extension for the *SURF* framework. It is also an essential module for our proposal due to its functions: score the alternative paths from the set proposed by the Bikes and Parking Prediction Module and generate arguments or incentives that will be presented to the user trying to slightly modify his/her route and behavior. Using these functions our system proposal acquires de capability of re-balancing the bike sharing service with a better redistribution of the bikes. Alternative routes are scored following a criteria which gives a higher score to those routes that are considered better for the service functioning. In this case the best alternative routes are those that ensure that the stations in the surroundings of user's preferred station maintain a balance between free bikes and free parking slots. This balance is crucial to the good functioning of the bike sharing service since tries to insure the availability of the resources. The re-balancing is user-driven since it is carried out by the users themselves at the moment they accept an alternative route. If a user agrees or accepts with the argumentation or incentives that this module has generated for any of the alternative departure/arrival station pair, then the user is moving the bikes according to our system rationale and helping to maintain a good distribution of the service resources. With this auto-balancing technique we can prevent user dissatisfaction that can be generated if the users are not able to find free bikes or free parking slots when they need one. This dissatisfaction can lead to a loss of service subscribers, which is a problem that bike sharing services tries to prevent, such as rise on the number of vehicles of individual and motorized vehicles. This increase arises other related problems among which we can find rise of the pollution levels, increase of the traffic congestion, etc.

6.1 Scoring

With the purpose of distinguishing the quality of the different proposals of alternative routes calculated by the the Bikes and Parking Prediction Module the system faces the need of being capable of scoring every alternative. Since each alternative route is composed of two stations (the departure station and the arrival station), first we need to score every station individually. Once the stations are evaluated, the next step is combining the score of both stations in a way that in the alternative route punctuations the balance of both stations is considered. The alternative route score will help to order the alternative routes by its optimality.

6.1.1. Station scoring

In order to evaluate every proposal, scoring the optimality of the stations is our first step. For this, we need to score each station individually disregarding if it is a departure station or an arrival. This restriction is needed due to the necessity of comparing or using the score obtained for both stations when scoring the alternative path as a whole. We use the next formulation for each station of the proposed pair:

$$score = \frac{\sum_{i=1}^{N} w_i o_i}{C}$$

$$\sum i = 1^N w_i = 1$$

$$t(o_1) < t(o_2) < ... < t(o_n)$$

Being:

- o_i the predicted station occupation on time i.
- w_i weights that controls the importance of the given occupation of the station the time instant i.
- **C** the maximum capacity of the station
- $t(o)$ a function that returns the date for a given occupation o.

This score measures the number of bikes that will be in the station weighted for the different time instants i. The higher the score the higher will be the estimated number of bikes in the station along the considered time frame. Since stations have different number of parking slots, the scores are normalized, being their possible values in the range [0,1]. Otherwise the stations with a higher number of parking slots will normally have higher score.This normalization allows us to compare the status between two stations independently of the number of parking slots they have.

The weights must be defined for each station at different times taking into account their behavior. Using them the algorithm can give more importance to the occupation at the estimated time when the user is going to make use of the resource (bikes or slots), to the occupation in a short term or to the occupation in a medium term. Hence, the weights contribute to better modeling the stations behavior or they simply can be used to modify the stations behavior and the user options if needed in specific cases.

Attending the case for departure stations, the optimal situation is given at the highest score. The stations with more available bikes are better scored propitiating that the user chooses a bike from those stations. This way we want to ensure that stations with bicycle shortage are more likely not to be completely empty, leading to the situation that they are not capable to offer this kind of resource. On the other hand, the stations with higher number of available bikes are more likely to be proposed as an alternative due to its higher score. This way we are preventing these stations from running out of free parking slots or, in the case of already full stations, freeing these resources in order to being able to continue offering them to the users. Concluding, working with departure stations the score should be maximized.

Dealing with the arrival stations case, we face an opposite situation that the given with the departure stations. A lower score means that the station holds more free parking slots. By suggesting first the stations with lower scores we ensure that empty or near empty stations will receive user's bike first. This way these stations will be able to offer bikes in the case they are being used as departure stations. Also, this prevents more occupied stations from receiving more bikes and keep the balance between available bikes and free parking slots.

In addition the formulations can be modified with some parameters that can represent the user preferences. As an example, the scores can be modified with a factor that represents the extra distance the user needs to travel with the alternative proposals. The same can be done with the time needed, the number of traffic lights, etc.

6.1.2. Alternative route scoring

Once the module has computed both scores (departure station score and arrival station score) it must combine them in order to score the alternative route as a whole. As already discussed, the departure station score should be maximized and the arrival station score should be minimized, our proposal for the alternative scoring is the following:

$$\text{alternative score} = w_d \text{ departure score} - w_a \text{ arrival score}$$

Where w_d and w_a are weights that determines the importance of the station in the given time frame. There are many several options when choosing the criterion to establish the weights: models of the users behavior or the stations states, the function of the station (departure station or arrival station) the system need to use an specific station, etc. Also, a combination of criteria can be done.

Once all the alternatives are scored the system can sort them from the best, which will contribute more to keep the bike sharing system balanced, to the worst. After sorting the alternatives the argumentation module has more information on which to base the arguments and incentives. Besides the score, the argumentation module uses information about the user such as preferences, history, etc. When the module finishes with all the alternatives it passes them including the arguments/incentives to the system manager in order to present them to the user.

6.2 Example scenario

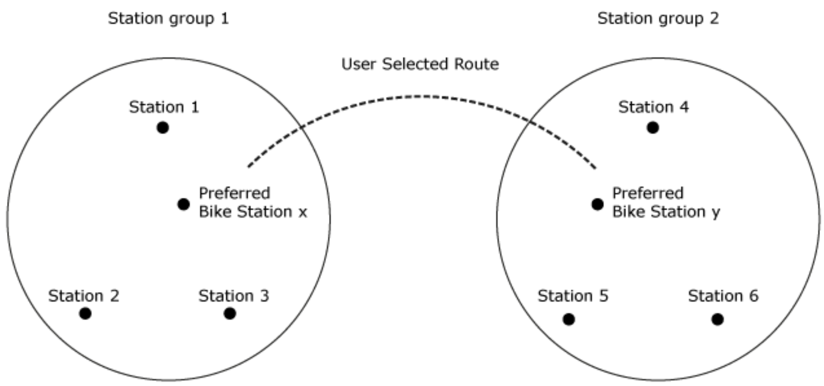


Figure 6.1: Example scenario

The example situation is represented in Figure 6.1. The user selects the station$PreferredBikeStation_x$ which is the departure station, and the station $PreferredBikeStation_y$ which is the arrival station. Following, the Bikes and Parking Availability Module predicts the occupation in some future time instants (one prediction minimum at 30 minutes from the estimated time in which the user arrives at the departure station). With the predictions, the Bikes and Parking Availability Module considers if the stations are suitable to be used without creating an undesired situation in the system or to the user i.e.: the departure station will be empty or the arrival station will be full. If is decided that the original stations can be used with no problems the process ends. If is decided that at least one of the stations can be problematic for an optimal use of the bike sharing service then the Bikes and Parking Availability Module will predict again for the time instants but this time for every pair in the Cartesian product of the group station 1 and the group station 2. Each group is formed applying some criteria to the preferred bike stations, for example the nearest stations.

Station	o1	o2	w1	w2	Score
1	19	18	0,3	0,7	0,915
2	10	19	0,3	0,7	0,815
3	19	10	0,3	0,7	0,635
4	15	15	0,3	0,7	0,75
5	0	0	0,3	0,7	0
6	20	20	0,3	0,7	1

Tabla 6.1: Station scoring example

Alternative	w1	w2	score
1-4	0,5	0,5	0,0825
1-5	0,5	0,5	0,4575
1-6	0,5	0,5	-0,0425
2-4	0,5	0,5	0,0325
2-5	0,5	0,5	0,4075
2-6	0,5	0,5	-0,0925
3-4	0,5	0,5	-0,0575
3-5	0,5	0,5	0,3175
3-6	0,5	0,5	-0,1825

Tabla 6.2: Alternative route scoring example 1

The resulting set of pair stations are the alternative routes. This set is sent to the efficient bike trip module that will evaluate each station as described on Algorithm 6.2 in order to evaluate every alternative route as indicated on Algorithm 6.3. To simplify the explanation we limit the occupation measurements to two time instants. In the case of the departure stations the occupations are selected from the moment the user makes the query and from the expected time the user will arrive to the station. In the arrival station the occupations are selected from the time the user starts it's travel and from the expected arrival time. In our example the weights occupation and results of the scoring the stations can be found at Table 6.1. The weights and results of scoring the alternative paths when the user wants to go from a station in the station group 1 to a station in the station group 2 are represented at Table 6.2. When the situation is the opposite, i.e. the departure station belongs to the station group 2 and the arrival station is in the station group 1, we obtain the scores represented at at Table 6.3.

As can be seen the re-balancing idea in which the algorithm is based works as expected. When the user needs to grab a bike from a station the system

Alternative	w1	w2	score
4-1	0,5	0,5	-0,0825
4-2	0,5	0,5	-0,0325
4-3	0,5	0,5	0,0575
5-1	0,5	0,5	-0,4575
5-2	0,5	0,5	-0,4075
5-3	0,5	0,5	-0,3175
6-1	0,5	0,5	0,0425
6-2	0,5	0,5	0,0925
6-3	0,5	0,5	0,1825

Tabla 6.3: Alternative route scoring example 2

tries to recommend the station pair that better balances the idea of trying to not empty or let at full capacity the stations. Because of this at the departure stations with more estimated bikes are obtaining a highest score meanwhile the situation in the arrival station is the opposite. If we compare the result that occur in both situations (when the departure station is in the group station 1 or in the group station 2), we can confirm that the higher a station score is in one situation the lower will be in the opposite. In the examples, as the weights are maintained the same for both situations, the scores are identical but with changed sign. This is due to the idea that a station that meets better conditions for being acceptable for the system in order to offer to the user a free bike is inherently bad when offering a free parking slot and vice versa.

6.3 Algorithms

Algorithm 1 Station scoring algorithm

Require: Station occupation set O, weights set W
Ensure: $score$

```
score = 0
for each o_i, w_i in O, W do
    score = score + o_i w_i
end for
```

Figure 6.2: Station scoring algorithm

Algorithm 2 Alternative route scoring algorithm

Require: Alternative pairs set A, weights set W
Ensure: Score set S

```
for each s_ai, s_di in A do
    score_di = station score (s_di)
    score_ai = station score (s_ai)
    score_Ai = w_di score_di − w_ai score_ai
    S ← score_Ai
end for
```

Figure 6.3: Alternative route scoring algorithm

CHAPTER 7
Conclusions

This work has presented a multi-agent system architecture to improve the efficiency of bike sharing systems. The main novelty of the approach comes from the introduction of user-driven balancing in the loop: attempting to persuade users to slightly deviate from their origin/destination stations, and balancing the system in the process. We expect that this architecture will help to provide a better service, increase user satisfaction, and optimize the management of the bike network by reducing the number of balancing operations carried out by the service providers' trucks.

We studied the state-of-the-art and the related work for the bike availability forecasting for shared bike systems and for the station balancing. Through this process we have acquired the necessary knowledge to understand the problem and to be able to propose solutions.

Collecting data was a crucial component of this work. We managed to obtain data from the bike sharing system of Valencia and we also collected weather conditions that can influence the system usage. The collected data is from a period of time long enough to have sufficient information about our use case when designing and implementing the necessary modules for the multi-agent system.

This collected data was analyzed in order to obtain a better understanding of the bike shared system. We were able to determine which data influence in the usage of the system for its later use in the Bikes and Availability Prediction Module. The records in the resulting dataset consisted of a station id, a timestamp decomposed into year, month , day, hour, minute, second, and weekday of the measurement, temperature, rainfall precipitation, wind speed, and the number of free parking slots and bikes available in the station. As mentioned in Chapter 4, we discarded two parameters from the dataset: season and events. Season data was discarded because the models can infer it from the date. Data from events was discarded due to the limited amount we were able to acquire.

The proposed architecture has two main components: an *efficient bike trip module* and a *bikes and parking availability prediction module*. The *bikes and parking availability prediction module* was designed with the aim of forecasting the occupation of the stations in the next 30 minutes due to the time restriction that the bike shared service imposes to the users. After designing the module we proceeded to its implementation. The prediction module uses a machine learning approach to estimate the foreseen bike station status based on real historic data of a given bike sharing service. The output of this module will be used by the *efficient bike trip module* to persuade the user to use the most appropriate stations according to the user preferences and the system balancing. In order to ensure the quality of the predictions we used two different machine learning algorithm for comparing reasons: Support Vector Regression (SVR) and Artificial Neural Networks (ANN). Results have shown that a ANN returned the best results for this predictor as was showcased in the experimental section of this work. Moreover the experiments have shown that this approach is feasible and accurate. The *efficient bike trip module* was designed with the aim of provide the multi-agent system the ability of self-balancing the number of parked bicycles in the stations in order to avoid unwanted situations such as empty stations or totally full stations. To achieve this the module uses the forecast realized by the the *bikes and parking availability module*. This information is processed by a set of algorithms that, with an heuristic approach, scores the stations and the alternative routes differencing which ones are better for the module objective: auto-balancing the stations.

CHAPTER 8
Future Work

There still exists some ideas that can be used in order to expand the proposed multi-agent system. The system can be improved including more information to the process. In our case study we only use information about the current status of the stations since is the only one provided. If we could reach an agreement with the service provider, it could be possible to obtain extra information such as travel information. With this new data we will be able to know where a bicycle is taken and where is left, obtaining usage patterns among other information.

Another possible expansion is to adapt the system to perform chain travels. Due to the time restriction imposed by the Valencia sharing-bike service provider, users are allowed to use bikes without extra charge only by 30 minutes. Since for some routes it may be necessary more time to complete the travel, users usually park the bike that are using and demand another to continue the travel. Considering this situation in the multi-agent system it will be capable of offering optimal routes in chain travels.

Regarding that the MAS is designed to be used by the bike-sharing system the natural reasoning is that they will be using it from a portable device such as a smartphone, table, etc. Building an application will let the user selecting stations, routes, receive notifications on status changes, etc. As smartphones and similar devices are equipped with GPS locators the use of the system will be easier for the user and the system will be able to collect geolocation data to keep improving its performance by generating models, user profiles, etc.

Bibliography

[1] Ayuntamiento de valencia: Estadisticas del area metropolitana. https://www.valencia.es, accessed: 2017-06-12

[2] Valenbisi gana usuarios. http://www.levante-emv.com/valencia/2017/04/29/valenbisi-gana-usuarios-primera-vez/1560137.html, accessed: 2017-06-12

[3] Wikipedia: Bike-sharing systems in numbers. https://en.wikipedia.org/wiki/Bicycle-sharing_system, accessed: 2017-07-12

[4] Wikipedia: List of bicycle-sharing systems. https://en.wikipedia.org/wiki/List_of_bicycle-sharing_systems, accessed: 2017-07-12

[5] Basak, D., Pal, S., Patranabis, D.C.: Support vector regression. Neural Information Processing-Letters and Reviews 11(10), 203–224 (2007)

[6] Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.F.: Route planning in transportation networks. In: Algorithm Engineering, pp. 19–80. Springer (2016)

[7] Bazzan, A.L., Klügl, F.: A review on agent-based technology for traffic and transportation. The Knowledge Engineering Review 29(03), 375–403 (2014)

[8] Billhardt, H., et al: Towards smart open dynamic fleets. Multi-Agent Systems and Agreement Technologies LNCS 9571, 410–424 (2016)

[9] Chhabra, R., Verma, S., Krishna, C.R.: A survey on driver behavior detection techniques for intelligent transportation systems. In: Cloud Computing, Data Science & Engineering-Confluence, 2017 7th International Conference on. pp. 36–41. IEEE (2017)

[10] Costa, A., Heras, S., Palanca, J., Jordán, J., Novais, P., Julián, V.: Argumentation schemes for events suggestion in an e-health platform. In: International Conference on Persuasive Technology. pp. 17–30. Springer (2017)

[11] Crespo, M.F.V.: Predicción de disponibilidad de bicicletas en estaciones de uso público (2014)

[12] Croci, E., Rossi, D.: Optimizing the position of bike sharing stations. the milan case. (2014)

[13] DeMaio, P.: Bike-sharing: History, impacts, models of provision, and future. Journal of Public Transportation 12(4), 3 (2009)

[14] Farahani, R.Z., Miandoabchi, E., Szeto, W., Rashidi, H.: A review of urban transportation network design problems. European Journal of Operational Research 229(2), 281 – 302 (2013), http://www.sciencedirect.com/science/article/pii/S0377221713000106

[15] Ghosh, S., Varakantham, P., Adulyasak: Incentivizing the use of bike trailers for dynamic repositioning in bike sharing systems (2017)

[16] Ghosh, S., Varakantham, P., Adulyasak: Robust repositioning to counter unpredictable demand in bike sharing systems (2017)

[17] Ghosh, S., Varakantham, P., Adulyasak, Y., Jaillet, P.: Dynamic repositioning to reduce lost demand in bike sharing systems. Journal of Artificial Intelligence Research 58, 387–430 (2017)

[18] Giret, A., Carrascosa, C., Julian, V., Rebollo, M.: A crowdsourcing approach for last mile delivery. Submitted to Transportation Research Part C: Emerging Technologies (2017)

[19] Hernández, E., Sanchez-Anguix, V., Julian, V., Palanca, J., Duque, N.: Rainfall prediction: A deep learning approach. In: International Conference on Hybrid Artificial Intelligence Systems. pp. 151–162. Springer (2016)

[20] Jovanović, A., Nikolić, M., Teodorović, D.: Area-wide urban traffic control: A bee colony optimization approach. Transportation Research Part C: Emerging Technologies 77, 329–350 (2017)

[21] Jr., J.B.: Finding the optimal locations for bike sharing stations: a case study within the city of Richmond, Virginia. Master's thesis, Rijksuniversiteit Groningen, the Netherlands (2003)

[22] Kull, M., Ferri, C., Martínez-Usó, A.: Bike rental and weather data across dozens of cities. In: ICML 2015 Workshop on Demand Forecasting (2015)

[23] Kwiecień, J., Pasieka, M.: Cockroach swarm optimization algorithm for travel planning. Entropy 19(5), 213 (2017)

[24] Li, Y., Zheng, Y., Zhang, H., Chen, L.: Traffic prediction in a bike-sharing system. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems. p. 33. ACM (2015)

[25] Lowalekar, M., Varakantham, P., Ghosh, S., Jena, S.D., Jaillet, P.: Online repositioning in bike sharing systems (2017)

[26] Ochando, L.C., Julián, C.I., Ochando, F.C., Ferri, C.: Airvlc: An application for real-time forecasting urban air pollution. In: Proceedings of the 2nd International Conference on Mining Urban Data-Volume 1392. pp. 72–79. CEUR-WS. org (2015)

[27] O'Mahony, E., Shmoys, D.B.: Data analysis and optimization for (citi)bike sharing. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 687–694. AAAI'15, AAAI Press (2015), http://dl.acm.org/citation.cfm?id=2887007.2887103

[28] Raviv, T., Kolka, O.: Optimal inventory management of a bike-sharing station. IIE Transactions 45(10), 1077–1093 (2013)

[29] Rebollo, M., Carrascosa, C., Julian, V.: Transport network analysis for smart open fleets. In: International Conference on Practical Applications of Agents and Multi-Agent Systems. pp. 433–444. Springer (2017)

[30] Rigas, E.S., Ramchurn, S.D., Bassiliades, N.: Managing electric vehicles in the smart grid using artificial intelligence: A survey. IEEE Transactions on Intelligent Transportation Systems 16(4), 1619–1635 (2015)

[31] Sanchez-Anguix, V., Aydogan, R., Julian, V., Jonker, C.: Unanimously acceptable agreements for negotiation teams in unpredictable domains. Electronic Commerce Research and Applications 13(4), 243 – 265 (2014), http://www.sciencedirect.com/science/article/pii/S1567422314000283

[32] Satunin, S., Babkin, E.: A multi-agent approach to intelligent transportation systems modeling with combinatorial auctions. Expert Systems with Applications 41(15), 6622–6633 (2014)

[33] Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks 61, 85–117 (2015)

[34] Schuijbroek, J., Hampshire, R., van Hoeve, W.J.: Inventory rebalancing and vehicle routing in bike sharing systems. European Journal of Operational Research 257(3), 992–1004 (2017)

[35] Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., Krause, A.: Incentivizing users for balancing bike sharing systems. In: AAAI. pp. 723–729 (2015)

[36] Teodorovic, D., Dell'Orco, M.: Bee colony optimization–a cooperative learning approach to complex transportation problems. Advanced OR and AI methods in transportation pp. 51–60 (2005)

[37] Teodorović, D., Dell'Orco, M.: Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization. Transportation Planning and Technology 31(2), 135–152 (2008)

[38] Ticknor, J.L.: A bayesian regularized artificial neural network for stock market forecasting. Expert Systems with Applications 40(14), 5501–5506 (2013)

[39] Yoon, J.W., Pinelli, F., Calabrese, F.: Cityride: a predictive bike sharing journey advisor. In: Mobile Data Management (MDM), 2012 IEEE 13th International Conference on. pp. 306–311. IEEE (2012)