Graduation Project Report

Topic:
AN ADAPTIVE LEARNING APPROACH FOR SHORT-TERM TAXI-PASSENGER DEMAND PREDICTION

Carried out by:
Amal Saadallah

Host Organization:
NEC Laboratories Europe

Supervisor: Dr. Luis Moreira-Matias
Pedagogical Advisor: Dr. Hassine Saidane
Address: Kurfürsten-Anlage 36, D-69115 Heidelberg, Germany
Phone: +49 6221 4342 235
Abstract

The increasing availability of GPS-equipped taxis is providing enormous and ever-growing trajectory data. This data plays a key role in building informed driving strategies which contribute to maintaining taxi companies sustainability. In this work, we focus on predicting short-term spatiotemporal distribution of taxi-passengers. Providing this knowledge to taxi-drivers contributes to solving the real-time choice problem of which is the best location to go after a passenger drop-off.

Keywords : ITS, Global Positioning System (GPS) data, Mobility Intelligence, Taxi-passenger demand, Machine Learning, Time-series forecasting, Time-varying Poisson models, Autoregressive integrated moving average (ARIMA), Vector AutoRegressive (VAR), Ensemble learning.

Résumé

La disponibilité croissante des taxis équipés de GPS fournit des quantités énormes de données de localisation. Ces données jouent un rôle clé dans l’élaboration des stratégies de conduite assistée, qui à leur tour contribuent au maintien de la durabilité des compagnies de taxi. Dans ce travail, nous nous concentrons sur la prévision de la distribution spatio-temporelle des passagers de taxi à court terme. Ces prévisions contribuent à la résolution de problème de choix du meilleur endroit pour trouver le prochain passager.

Keywords : ITS, Géo-positionnement par Satellite (GPS), Apprentissage Automatique, Prévision des séries temporelles, Processus de Poisson non-homogènes, Modèle Autorégressif (ARIMA), Vecteur Autorégressif (VAR), Ensemble d’Apprentissage.
To my parents, sine qua non
Acknowledgements

I owe my heartfelt gratitude to my supervisor Dr. Luis Moreira-Matias, for the prodigious opportunity he gave me to work with his highly regarded team of research scientists and engineers. I’m thankful for his confidence, his precious guidance, his valuable suggestions, his continuous support and his constant availability which brought this project to light.

I’m also grateful to all the staff of the Intelligent Transportation Systems division at NEC especially Jihed Khiari who contributed to the success of this project. I would also like to take the opportunity to thank my teachers at Tunisia Polytechnic School for considerably contributing to my professional and academic development as well as my supervisor Dr. Hassine Saidane.

Finally, I must express my very profound gratitude to my parents and to my sisters for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

This accomplishment would not have been possible without you all. Thank you.
# Contents

General Introduction 2

1 Overview 3
   1.1 Presentation of the Host Organization ................................. 3
      1.1.1 NEC Laboratories Europe ........................................ 3
      1.1.2 Intelligent Transportation Systems Group ......................... 4
   1.2 General Context of the Project ........................................ 4

2 Key Concepts and State of the Art 8
   2.1 Key Concepts .......................................................... 8
      2.1.1 Time series Basics ................................................ 8
         2.1.1.1 Components of a time series ................................. 9
         2.1.1.2 Stationary time series ..................................... 9
      2.1.2 Data analysis basics .............................................. 10
      2.1.3 Global Positioning System (GPS) Data .......................... 11
   2.1.4 State of the Art ................................................... 12
      2.1.4.1 Time Varying Poisson Model ................................. 13
      2.1.4.2 Weighted Time Varying Poisson Model ....................... 14
      2.1.4.3 AutoRegressive Integrated Moving Average Model .......... 14

3 Methodology 16
   3.1 A drift-aware VAR model ............................................. 16

Graduation Project Report iv
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.1 The vector autoregressive (VAR) model</td>
<td>16</td>
</tr>
<tr>
<td>3.1.2 Cross-correlation function</td>
<td>17</td>
</tr>
<tr>
<td>3.1.3 Stands Selection Criteria</td>
<td>18</td>
</tr>
<tr>
<td>3.1.4 Hoeffding-Bound Test</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Best Real-time Induction of Generic and Heterogeneous short-term Taxi-passenger demand Framework</td>
<td>21</td>
</tr>
<tr>
<td>3.2.1 Model Selection</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2 Page-Hinkley Test</td>
<td>22</td>
</tr>
<tr>
<td>3.2.3 Sliding-Window Ensemble</td>
<td>23</td>
</tr>
<tr>
<td>4 Case Studies: Data Acquisition and Preprocessing</td>
<td>26</td>
</tr>
<tr>
<td>4.1 Case Study A:</td>
<td>26</td>
</tr>
<tr>
<td>4.1.1 Data Acquisition and Preprocessing</td>
<td>26</td>
</tr>
<tr>
<td>4.1.2 Data Analysis</td>
<td>28</td>
</tr>
<tr>
<td>4.2 Case Study B:</td>
<td>30</td>
</tr>
<tr>
<td>4.2.1 Data Acquisition and Preprocessing</td>
<td>30</td>
</tr>
<tr>
<td>4.2.2 Data Analysis</td>
<td>31</td>
</tr>
<tr>
<td>5 Experimental Results</td>
<td>33</td>
</tr>
<tr>
<td>5.1 Experiments</td>
<td>33</td>
</tr>
<tr>
<td>5.1.1 Hardware Environment</td>
<td>33</td>
</tr>
<tr>
<td>5.1.2 Software Environment</td>
<td>34</td>
</tr>
<tr>
<td>5.1.3 Experimental Setup</td>
<td>35</td>
</tr>
<tr>
<td>5.2 Evaluation Metrics</td>
<td>37</td>
</tr>
<tr>
<td>5.3 Results and Discussion</td>
<td>38</td>
</tr>
<tr>
<td>5.3.1 Results</td>
<td>39</td>
</tr>
<tr>
<td>5.3.2 Discussion</td>
<td>41</td>
</tr>
<tr>
<td>Conclusion</td>
<td>47</td>
</tr>
</tbody>
</table>
List of Figures

1.1 NEC Smart Transportation [1] .................................................. 5
2.1 Real-time GPS Vehicle Tracking System ............................... 12
3.1 An example of $ccf$ plot between two times series .................... 18
3.2 BRIGHT Framework ............................................................ 24
4.1 Taxi-stand spatial distribution in the city of Porto, Portugal ........ 27
4.2 Frequency Distribution of Taxi Cruise Time for the entire data set of Porto, Portugal .................................................. 29
4.3 Taxi-stand spatial distribution in the city of Thessaloníki, Greece .... 31
5.1 Illustration of the stand selection process for 24h period for the stand 11, Porto ................................................................. 41
5.2 Illustration of Performance of the Page-Hinkley(PH)-test ............ 44
5.3 BRIGHT evaluation on a typical work day (Porto) .................... 45
5.4 Comparison between BRIGHT and S.o.A. Ensemble on a typical work day (Porto) .......................................................... 45
5.5 Weekly comparison between the forecasted services and the real-emerged services on two distinct scenarios/taxi stands and weeks .... 46
List of Tables

4.1 Taxi Services Volume (Per Daytype/Shift) .......................... 30
4.2 Taxi Services Volume (Per Driver) .............................. 30
4.3 Taxi Services Volume (Per Daytype/Shift) .......................... 32
4.4 Taxi Services Volume (Per Driver) .............................. 32

5.1 Description of the Learning Periods ................................. 37
5.2 Error Measured on The Models Using sMAPE and NRMSE .......... 40
General Introduction

Taxi is one of the major transportation modes due to its personalized, direct and efficient point-to-point service. Matching taxi supply with demand is one of the biggest challenges faced by taxi companies today. The unbalance between demand and supply results in two possible scenarios: Scenario 1, i.e., an excess in vacant vehicles and taxi drivers wasting time and fuel wandering around the streets or waiting in stands for a passenger and Scenario 2, i.e., larger waiting time for passengers and lower taxi reliability. Solving this issue enables not only to enhance both taxi companies’ profitability and reliability, but also to reduce traffic pollution and emissions. To approach this problem, a fundamental question should be raised: How can we achieve a near-optimal taxi distribution over time to meet the demand?

Providing knowledge about the spatiotemporal distribution of passengers is a key feature in achieving efficient taxi dispatching over time and space. As more and more taxis are equipped with GPS sensors and wireless communication units, enormous trajectory and taxi status data is collected in real-time. This data plays pivotal role in this topic because it has been proven to be extremely useful for unlocking knowledge about traffic dynamics of a city’s road network [4, 8, 13] and human mobility patterns [3, 10, 11, 12].

Since GPS data is mainly a live data stream, we make use in this work of this main advantage to build a real-time prediction framework for the number of services that will emerge at a given taxi stand in a short-time horizon of $P$-minutes. Our framework contributes to solving the real-time choice problem about which is the best taxi stand
to go to after a passenger drop-off.

We tested our framework by choosing two large-sized taxi fleets running in two different cities, as case studies. In the case study A), we considered a fleet of 438 vehicles with 63 stands in the city of Porto, Portugal. In the second case B), we had a fleet of 1295 taxis running in the city of Thessaloníki, Greece. The city contains a total of 68 stands.

The remainder of this report is structured as follows. In Chapter 1, we will give an overview of the project. Chapter 2 presents the definition of the key concepts and a reminder of the state of the art. In chapter 3, we will thoroughly detail our methodology. Chapter 4 is dedicated to describe data acquisition and preprocessing for both case studies. In chapter 5, we will introduce the experimental setup and the used metrics to evaluate our framework; then, the obtained results are detailed. Finally, conclusions are drawn as well as future works.
Overview

Throughout this chapter, we will present the general context of the project. We will start by presenting the host organization which proposed and accommodated it.

1.1 Presentation of the Host Organization

1.1.1 NEC Laboratories Europe

NEC Corporation is a Japanese multinational provider of information technology (IT) services and products, with its headquarters in Minato, Tokyo, Japan. It is a leader in the integration of IT and network technologies that benefit businesses and people around the world. By providing a combination of products and solutions that cross utilize the company’s experience and global resources, NEC’s advanced technologies meet the complex and ever-changing needs of its customers. Bringing more than 100 years of expertise in technological innovation, NEC is focused on creating Smart City solutions that help build an efficient and sustainable infrastructure.

NEC Europe Ltd. is a subsidiary of NEC Corporation and has its headquarters in South Ruislip, UK. In 1994, NEC Europe Ltd established a research and development division called NEC Laboratories Europe. NEC Laboratories Europe (NLE) is located in Heidelberg, Germany. NLE brings together more than a 100 leading researchers from all over the world and holds close ties with leading European research institutes and universities. NLE also collaborates with major industries in Europe, e.g. network
operators, ICT vendors, automotive companies, utilities, providers, etc. NLE conducts research and development on cutting-edge ICT technologies, in particular:

- 5G, Future Internet, incl. SDN
- Cloud Platform, management and service
- Internet-of-Things (M2M) platform and services
- Security, privacy and performance
- Smart Energy
- Intelligent Transportation Systems

1.1.2 Intelligent Transportation Systems Group

NLE’s Intelligent Transportation System (ITS) group established a leading position in 5.9 GHz communication units for Cooperative Systems, including standardization, pilots and its award-winning experimental platform. NLE has also been working on transport optimization as well as transportation data analytics by applying big data technologies.

Being a leading machine learning and data analytics platform vendor, as well as an ITS supplier and system integrator, NLE-ITS is in a key position to apply data analytics technologies to transportation. In particular, it is engaging with cities, automotive industries and transport operators to exploit the potential of big data to improve and enable Smart Cities components like autonomous driving-based services, mobility demand prediction, bus scheduling optimization and more. The figure below shows the pillars of the group vision of Intelligent Transport Systems.

1.2 General Context of the Project

The increasing availability of GPS-equipped taxis is providing continuous and enormous trajectory data[3, 4]. This data is one of the main variables contributing to the
development of intelligent transportation systems such as efficient taxi dispatching systems[6], time-saving route finding [7, 8], fuel saving routing[9]...

However, taxi companies and drivers still facing real challenges because of the imbalance between taxi supply and customer demand over time and space. The imbalance between demand and supply results in two possible scenarios: Scenario 1, i.e., an excess in vacant vehicles and taxi drivers wasting time and fuel wandering around the streets or waiting in stands for a passenger and Scenario 2, i.e., larger waiting times for passengers and lower taxi reliability.

Providing knowledge about the spatiotemporal distribution of passengers is a key feature in achieving efficient taxi dispatching and recommending efficient cruising strategies for taxi drivers to find their next passenger (especially when there is no economic viability of adopting random ones). As more and more taxis are equipped with GPS sensors and wireless communication units, enormous trajectory and taxi status data is collected in real-time. This data plays pivotal role in this topic because it has been proven to be extremely useful for unlocking knowledge about traffic dynamics of a city’s road network [4, 8, 13] and human mobility patterns [3, 10, 11, 12]. So far,
multiple works have explored this type of data successfully with various applications such as efficient taxi dispatching systems[6], passenger-finding strategies[15, 14], time-saving route finding [7, 8], fuel saving routing[9], amongst others.

In this work we build a real-time prediction framework for the number of services that will emerge at a given taxi stand in a time horizon of $P$ minutes. Our framework contributes to solve the real-time choice problem about which is the best taxi stand to go to after a passenger drop-off. An intelligent approach regarding this problem will improve taxi services reliability and profitability since it helps to establish an intelligent distribution of vehicles throughout stands which will reduce the average waiting time at stands to pick up a passenger, and the vacant cruising miles searching for passengers in the streets while potential demand may exist in some near stands. Furthermore, passengers will also experience a lower waiting time to get a vacant taxi (automatically dispatched or directly picked up at a stand). Aside from the number of emerged services, the stand choice problem involves many other variables such as the distance/cost relation with each stand, the number of taxis already waiting at each stand… Traffic and taxi network sensors can always provide information about the mentioned variables. However, the work described here will just focus on passenger demand prediction over space (i.e. taxi stand) for a short-term horizon of $P$-minutes.

One of the most recent advances on this topic was presented by Moreira-Matias et. al in [22]. For the same purpose, they proposed a sliding-window ensemble framework that handles three distinct types of memory. Their framework is able to build accurate predictions on a stream environment and to update itself to learn from the novelty thereby introduced. However, it presents two relevant limitations:

- It uses two similar non-homogeneous Poisson models out of three predictive models. Both models’ weights and memory size are static. The unique difference between the two models is the length of memory they handle to build predictions. This results in a biased ensemble framework;

- It assumes the absence of dependencies between the demand at different taxi stands over time while the fluctuations of the demand at a given taxi stand can be attributed to the fluctuations of the demand at one or multiple other taxi stands;
We propose a way to minimize the previously described limitations as much as possible by:

- Exploiting the potential existing dependencies between the demand at different taxi stands over time to build predictions using a multivariate time series model, the vector autoregressive (VAR);

- Continuously monitoring changes in those dependencies over time to build a drift-aware VAR model;

- Reducing bias in the proposed framework by performing a prior selection between each two similar models. The resulting models are introduced in a sliding-window ensemble model, the same used in [22];
Chapter 2

Key Concepts and State of the Art

We will introduce in this section some useful definitions and concepts that shed light on the project. We will also give an overview of the existing state-of-the-art.

2.1 Key Concepts

We define in this section the key concepts necessary to put our project in its context.

2.1.1 Time series Basics

A time series is a sequence of measurements of one variable collected over time. Those measurements are made at regular time intervals. When dealing with time series, some important questions need to be answered:

- Is there a trend, which means if the time series tend to increase (or decrease) on average over time?
- Is there seasonality, Is there some repeating patterns of highs and lows related to a time period (hours/ day/month...), and so on?
- Are there outliers? With time series data, outliers are data points which are far away from your other observed data.
- Is the series stationary or not?
2.1.1.1 Components of a time series

Most of time series can be described with three terms representing three different aspects: *seasonality* component $S_t$, *trend-cycle* component $T_t$ and a *remainder* (error) component $E_t$.

\[ x_t = f(S_t, T_t, E_t) \] (2.1)

**Trend pattern** A trend is observed when we can notice a long-term increase/decrease in the data. So, the trend is referred to changing direction and it might be an increasing trend or a decreasing trend.

**Seasonal pattern** When there are patterns that repeat over known, fixed periods of time within the data set, such patterns are known as seasonality, seasonal variation, periodic variation, or periodic fluctuations. This variation can be either regular or semi-regular. A seasonal pattern exists when a series is influenced by seasonal factors (the day of the week, the month of the year, the day of the month, etc.). The seasonality is formally defined as a correlation dependency of order $k$ called the lag between each $i^{th}$ element of the series and the $(i - k)^{th}$ element and measured by the autocorrelation function. If the measurement error is not too large, seasonality can be graphically identified as a pattern that repeats itself every $k$ elements.

**Cyclic pattern** A cyclic pattern is identified when some rises and/or falls are observed, but not with a fixed period. If the observed fluctuations in data are not attributed to a fixed time period, then they are cyclic. Otherwise, the data pattern is seasonal.

2.1.1.2 Stationary time series

In cases where the stationary criterion are violated, the first requisite becomes to stationarize the time series and then try stochastic models to predict this time series. There are multiple ways of stationarizing the time series. Some of them are Detrending, Differencing etc. There are three basic criterion for a series to be classified as stationary time series:

- The mean of the series should not be a function of time rather should be a con-
- The variance of the series should be also constant. This property is known as homoscedasticity.

- The covariance of the \(i\)th term and the \((i + m)\)th term should not be a function of time.

One of the most commonly used statistical tests to check the stationarity of time series is the Dickey Fuller test. Stationarity testing and converting a series into a stationary series are the most critical processes in a time series modeling.

### 2.1.2 Data analysis basics

Data analysis is the science of examining, cleaning, transforming and modeling data with the purpose of unlocking knowledge, discovering useful information, using this knowledge, drawing conclusions and supporting decision making. The process of data analysis includes several steps. Every step should be fulfilled in order to get the required information. These steps are detailed as follows:

**Data requirements:** what to measure and for which purpose?

**Data collecting:** once the data of interest is determined, we can find out whether it can be collected from the existing databases or whether we should look for new sources.

**Data processing:** data initially obtained should be organized for analysis (Example: placing data into rows and columns in a table format for further analysis)

**Data cleaning:** The collected data can be unready for analysis and modeling directly. Like it can be incomplete. For example, it may contain some duplicate values, missing rows or values, outliers/noise, etc... Data cleaning consists in properly addressing these issues.

**Exploratory data analysis:** once the data has been collected, processed and cleaned, it is now ready for analysis. Many techniques can be applied to understand the messages transferred by this data. In addition, descriptive statistics can be generated to better understand the data structure.
Data Visualizing: data visualization techniques usually help to clearly and efficiently understand data and they are also useful for communicating the message to the audience. Data visualization presents the information in different formats such as tables and charts to help communicate important information contained in the data.

Modeling and algorithms: mathematical algorithms may be applied to the data in order to identify relationships between the variables, such as correlation or causation.

### 2.1.3 Global Positioning System (GPS) Data

The Global Positioning System (GPS) is a satellite-based system that can be used to locate positions anywhere on the earth. Operated by the U.S. Department of Defense (DoD), NAVSTAR (NAVigation Satellite Timing and Ranging) GPS provides continuous (24 hours/day), real-time, 3-dimensional positioning, navigation and timing worldwide. Any person with a GPS receiver can access the system, and it can be used for any application that requires location coordinates.

The GPS system consists of three segments: 1) The space segment: the GPS satellites themselves, 2) The control system, operated by the U.S. military, and 3) The user segment, which includes both military and civilian users and their GPS equipment.

The use of GPS covers many domains. In transportation, Automatic Vehicle Location (AVL) data which determines the exact location of vehicles in real-time, is based on GPS. This data is transmitted to a data server as shown in Figure 2.1, where it is stored for later preprocessing and mining. Given the availability and accuracy of this data, many transport operators have equipped their fleets with on-board transmitters to collect it. Nowadays, more and more taxis are equipped with GPS sensors and wireless communication units. Enormous trajectory and taxi status data is collected in real-time. The GPS historical data play pivotal role in this topic because it can unlock the knowledge of human mobility patterns and social functional regions[10, 11, 12, 13]. Multiple works in the literature have already explored this type of data and came up with multiple applications such as passenger-finding strategies[15, 14] modeling passenger demand to provide efficient taxi dispatching methods[18, 19], uncovering taxi drivers’ cruising behavior patterns [16, 17] to afford better understanding of high-level...
Figure 2.1: Real-time GPS Vehicule Tracking System

human behavior and mobility intelligence. Although GPS data is mainly a live data stream, few researches made use of this main advantage to provide live information about passenger/taxi location in a specific date/time[22, 21].

2.1.4 State of the Art

One of the most recent advances on this topic was presented by Moreira-Matias et. al in [22]. They proposed a framework to predict at the instant t how many services will emerge during the future period \([t; t + P]\) at each existent taxi stand, reusing the information constantly transmitted/received by the telematics installed in each taxi about the current period (i.e. the framework runs continuously in a stream). First, the information was aggregated into a histogram time series. Then, three time-series forecasting techniques, respectively Time-Varying Poisson, Weighted Time-Varying Poisson and AutoRegressive Integrated Moving Average (ARIMA), were combined to generate accurate predictions on a stream environment. In this section, we will formally describe the three forecasting techniques used in the aforementioned work. Let \(X_{k,p} = (X_{k,t-1}, \ldots, X_{k,t-p+1}, X_{k,t-p})\) be a vector of the \(p\) past observations (discrete time series with aggregation period of \(P\)-minutes) for the demand at the stand \(k\).
2.1.4.1 Time Varying Poisson Model

This model is based on considering the probability of emergence of n taxi services in a determined time period - P(n) - following a Poisson Distribution. We can define it using the following equation

\[ P(n; \lambda) = \frac{e^{-\lambda} \lambda^n}{n!} \]  

(2.2)

where \( \lambda \) represents the rate (averaged number of the demand on taxi services) in a fixed time interval. However, in this specific problem, the rate \( \lambda \) is not constant but time-variant. So, it needs to be formulated as a function of time. Let \( \lambda_0 \) be the average (i.e. expected) rate of the Poisson process over a full week. \( \lambda(t) \) can be defined as follows

\[ \lambda(t) = \lambda_0 \delta_d(t) \eta_{d(t), h(t)} \]  

(2.3)

where \( \delta_d(t) \) is the relative change for the weekday \( d(t) \) (e.g.: Saturdays present lower day rates than Tuesdays); \( \eta_{d(t), h(t)} \) is the relative change for the period \( h(t) \) on the day \( d(t) \) (e.g. the peak hours); \( d(t) \) represents the weekday 1=Sunday, 2=Monday, ...; and \( h(t) \) indicates the interval (e.g., \( P \) minutes periods) in which time \( t \) falls (e.g. the time 00:31 is contained in period 2 if we consider 30-minutes periods). Consider \( \lambda(t) \) to be a discrete function (e.g.: an histogram time series of event’ counts aggregated in periods of \( P \) minutes). The equation 5.4 requires the validity of both equations:

\[ \sum_{i=1}^{7} \delta_i = 7 \]  

(2.4)

\[ \sum_{i=1}^{I} \eta_{d,i} = I, \forall d \]  

(2.5)

where \( I \) is the number of time intervals in a day. The result is discrete time series per stand representing the expected demand during an entire week: \( \lambda(t)_k \). Each value in this series is an average of all demands previously measured in the same day-type and period (i.e. the expected service demand for a Monday from 8:00 to 8:30 is the average of the demand on all past Mondays from 8:00 to 8:30).
2.1.4.2 Weighted Time Varying Poisson Model

The above model presented can be considered as a time-dependent average which generates predictions based on long-term memory size data (i.e. we consider all the historical records of the demand). However, it is not guaranteed that each taxi stand will have a highly regular demand: in fact, the demanded services in many stands can often be seasonal. To model this effect, a weighted average model is introduced. Its definition is based on the model presented previously: the goal of introducing weights is to increase the relevance of the demand pattern observed in the recent periods (i.e. recent weeks) (e.g. what happened on the previous Monday is more relevant than what happened two or three Mondays ago). The weight set \( \omega \) is calculated using a well-known time series approach to these type of problems: the Exponential Smoothing [25]. It is possible to define \( \omega \) as follows:

\[
\omega = \alpha \times (1, (1 - \alpha), (1 - \alpha)^2, \ldots, (1 - \alpha)^{\gamma - 1}), \gamma \in \mathbb{N}
\]  

(2.6)

where \( \gamma \) is the number of historical periods considered and \( 0 < \alpha < 1 \) is the smoothing factor (i.e. \( \gamma \) and \( \alpha \) are user-defined parameters). Then, based on the previous definition of \( \lambda(t)_k \), it is possible to define the resulting weighted average \( \mu(t)_k \) as follows:

\[
\mu(t)_k = \sum_{i=1}^{\gamma} \frac{X_{t-(\theta+x)} \times \omega_i}{\Omega}, \Omega = \sum_{i=1}^{\gamma} \omega_i
\]  

(2.7)

where \( \theta \) is the number of time periods contained in a week.

2.1.4.3 AutoRegressive Integrated Moving Average Model

In time series analysis, an AutoRegressive Integrated Moving Average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied to stationarize the time series [2].
The AR part of ARIMA indicates that the evolving variable of interest is regressed on its own lagged (i.e., prior) values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past. The I (for "Integrated") indicates that the data values have been replaced with the difference between their values and the previous values (and this differencing process may have been performed more than once). The purpose of each of these features is to make the model fit the data as well as possible.

Non-seasonal ARIMA models are generally denoted ARIMA\( (p, d, q) \) where parameters \( p \), \( d \), and \( q \) are non-negative integers, \( p \) is the order (number of time lags) of the autoregressive model, \( d \) is the degree of differencing (the number of times the data have had past values subtracted), and \( q \) is the order of the moving-average model.

We can formulate the previous process that generates the time series prediction (taxi demand over time for a given stand \( k \)) as:

\[
R_{k,t} = k_0 + \phi X_{k,p}^T + \theta \epsilon_{k,q}^T
\]

where \( R_{k,t} \) is the actual value at time period \( t \) and \( \epsilon_{k,q} = (\epsilon_{k,t}, \epsilon_{k,t-1}, \epsilon_{k,t-2}, \ldots, \epsilon_{k,t-q}) \) a vector of the Gaussian white noise' error terms observed in the past signal; \( \phi = (\phi_1, \phi_2, \ldots, \phi_p) \) and \( \theta = (1, \theta_1, \theta_2, \ldots, \theta_q) \) are respectively \( (1 \times p) \) and \( (1 \times (q + 1)) \) vectors representing the model parameters/weights while \( p \) and \( q \) are positive integers often referred to the order of the model. Both order and weights can be inferred from the historical time series using both the autocorrelation and partial autocorrelation functions as proposed by Box and Jenkins in [27].

**Conclusion**

This chapter has introduced the key concepts that our project explores. We have defined times series and data analysis basics since our work will be focusing on time series forecasting. Furthermore, we have described the main source of the available data. Finally, we have formally presented the time series forecasting methods used in the state of the art.
Chapter 3

Methodology

We will present in this section an overview of our framework. We will then detail each of the steps we followed. Our goal is to build a framework, which determines at the instant \( t \) the number of services that will emerge in the next \( P \) minutes at different taxi stands.

The models that have been used so far, are univariate time series models (i.e. generate short-term taxi demand prediction using only the data stream of the target stand as input). Although the use of univariate time series models can be very useful to provide sound predictions, there may also exist some cases where the specification of univariate time series models is hampered by fluctuations that can be attributed to one or multiple time series other than the target one. In our case, the demand at one particular taxi stand \( S_i \) can be affected by the demand generated over time by some or all the other stands in the same city or area. To approach this problem, we introduced a widely used multivariate time series model, The Vector Autoregressive VAR with the three existing models.

3.1 A drift-aware VAR model

3.1.1 The vector autoregressive (VAR) model

The vector autoregressive (VAR) model is one of the most commonly used multivariate time series models due to its灵活性 and easy specification[23]. It is a natural
extension of the univariate autoregressive model to dynamic multivariate time series. The VAR model has proven to be especially useful for describing the dynamic behavior of time series and for forecasting. Let \( S = (S_1, S_2, ..., S_N) \) be the set of the \( N \) taxi stands of interest and \( X = (X_{1,t}, X_{2,t}, ..., X_{N,t})^T \)  \((N \times 1)\) vector of time series variables for the demand at the \( N \) stands at the instant \( t \). VAR model describes the evolution of a set of \( N \) variables at the instant \( t \) as a linear function of their past values. The basic \( p \)-lag vector autoregressive (VAR(\( p \))) model has the following form:

\[
\begin{pmatrix}
X_{1,t} \\
X_{2,t} \\
\vdots \\
X_{N,t}
\end{pmatrix}
= \begin{pmatrix} c_1 \\
c_2 \\
\vdots \\
c_N
\end{pmatrix} + \sum_{i=1}^{p} \begin{pmatrix}
\pi_{11}^i & \pi_{12}^i & \cdots & \pi_{1N}^i \\
\pi_{21}^i & \pi_{22}^i & \cdots & \pi_{2N}^i \\
\vdots & \vdots & \ddots & \vdots \\
\pi_{N1}^i & \pi_{N2}^i & \cdots & \pi_{NN}^i
\end{pmatrix} \begin{pmatrix}
X_{1,t-i} \\
X_{2,t-i} \\
\vdots \\
X_{N,t-i}
\end{pmatrix} + \begin{pmatrix}
\epsilon_{1,t} \\
\epsilon_{2,t} \\
\vdots \\
\epsilon_{N,t}
\end{pmatrix}
\]  

Where \( \Pi_i = (\pi_{ij}^i) \in \mathbb{R}^{N \times N} \) are coefficient matrices and \( \epsilon_t \) is an \((N \times 1)\) vector of an unobservable zero mean white noise process. VAR model building including specification and estimation steps can be conducted using the iterated procedure proposed by Box and Jenkins [27]. The two previous steps can be found with details in [23]. For VAR models, the model specification is to identify the lag \( p \), called also VAR order. Many approaches have been presented in the literature to identify the VAR order. We can mention the sequential likelihood ratio tests and the information criteria method [23]. In our work, we use the cross-correlation(ccf) functions for the order selection.

### 3.1.2 Cross-correlation function

The cross-correlation function (ccf) of two time series is the product-moment correlation as a function of lag, or time-offset, between the series. Consider \( N \) pairs of observations on two time series of the demand at two different taxi stands \( S_i \) and \( S_j \), \( X_i \) and \( X_j \) respectively. We can define the ccf using the cross-covariance function (ccvf) given by:

\[
c_{X_i X_j}(k) = \frac{1}{N} \sum_{t=1-k}^{N} (X_{i,t} - \bar{X}_i)(X_{j,t+k} - \bar{X}_j), [k = -1, -2, ..., -(N-1)]
\]  

Graduation Project Report
where $N$ is the series length, $\bar{X}_i$ and $\bar{X}_j$ are the sample means, and $k$ is the lag. The sample cross-correlation function ($ccf$) is the $ccvf$ scaled by the variances of the two series:

$$
ccf_{X_i,X_j}(k) = \frac{c_{X_i,X_j}(k)}{\sqrt{c_{X_i,X_i}(0)c_{X_j,X_j}(0)}}
$$

(3.3)

Where $c_{X_i,X_i}(0)$ and $c_{X_j,X_j}(0)$ are the sample variances of $X_i$ and $X_j$.

The equation 3.2 describes the situation and summarizes lagged correlations where $X_j$ “leads” $X_i$, $X_i$ “lags” $X_j$. The sample cross correlation function ($ccf$) is helpful for identifying lags of the $X_j$-variable that might be useful predictors of $X_i$. The Figure 3.1 shows an example of cross correlation function ($ccf$) plot. It enables a comparison between ccf values for different lags and a user-defined significance level.

![Figure 3.1: An example of ccf plot between two times series](image)

### 3.1.3 Stands Selection Criteria

A naive way of using VAR model in our context to build predictions is to follow the above formulation which considers as input for VAR model the historical counts of dispatched services from all the stands and generates as output $(N \times 1)$ vector of the predicted number of emerged services at the $N$ stands. It is more reasonable to select for each stand $S_i$ the subset of stands (instead of all the stands) which their times series
of the demand are more relevant for predicting the demand at $S_i$. The choice of this subset should be based on the existing dependencies between the demand at $S_i$ and the demand at the other stands. Indeed, some research questions arise from this analysis: How can we evaluate the dependencies between the stands over time and use them to select the optimal subset of stands for each target stand? However, dependencies between the stands may change over time. Those changes can be attributed to the dependencies structure itself or to some abrupt changes in the traffic dynamics due to some circumstances. So, how can we detect significant changes in those dependencies that requires some updates in the initially selected subset of stands for each stand?

To approach the raised issues, we set two selection criteria for each stand $S_i$ to choose the subset of stands that will include their number of services over time as features to predict future demand at $S_i$ and we continuously monitor the dependencies between the stands to introduce the appropriate updates to the selected set of stands. In this work, we use the temporal correlation as a distance metric between the disposed time series (number of services at different taxi stands). Based on this distance, the first selection criteria is a distance threshold that is defined as the maximum of distance between the target stand and the other stands (i.e. if the distance between the target stand $S_i$ and another stand $S_j, j \neq i$, is superior to this threshold, $S_j$ will not be selected). The second selection criteria is defined as the optimal number $k$ of stands whose demand data stream has the most contribution to the demand at $S_i$, called later top-$k$ stands. The top-$k$ stands and the distance threshold are determined through a hyper-parameters tuning process. So, given a stand $S_i$, after selecting the appropriate set of top-$k$ stands, we can formulate the above process that generates the demand at $S_i$ at the instant $t$ as follows:

Let $S_k = (S_{j1}, S_{j2}, ..., S_{jk})$ the set of the top-$k$ stands:
\[
\begin{pmatrix}
X_{j1,t} \\
\vdots \\
X_{i,t} \\
\vdots \\
X_{jk,t}
\end{pmatrix}
= 
\begin{pmatrix}
c_{j1} \\
\vdots \\
c_i \\
\vdots \\
c_{jk}
\end{pmatrix}
+ \sum_{i=1}^{p}
\begin{pmatrix}
\pi_{j1j1}^i & \pi_{j1j2}^i & \cdots & \pi_{j1jk}^i \\
\vdots & \vdots & \ddots & \vdots \\
\pi_{ij1}^i & \pi_{ij2}^i & \cdots & \pi_{ijk}^i \\
\vdots & \vdots & \ddots & \vdots \\
\pi_{jk1}^i & \pi_{jk2}^i & \cdots & \pi_{jkjk}^i
\end{pmatrix}
\begin{pmatrix}
X_{j1,t-i} \\
\vdots \\
X_{i,t-i} \\
\vdots \\
X_{jk,t-i}
\end{pmatrix}
+ 
\begin{pmatrix}
\epsilon_{j1,t} \\
\vdots \\
\epsilon_i,t \\
\vdots \\
\epsilon_{jk,t}
\end{pmatrix}
\tag{3.4}
\]

Although the hyper-parameters tuning is performed periodically, some changes in the dependencies between the stands may occur suddenly because of an abrupt change in the demand at some stands that may last for short or long period. To face this phenomenon, we introduce a change detection algorithm based on Hoeffding-Bound test to monitor significant changes in the dependencies between the stands over time, more precisely, in the temporal correlation between the stands for each time prediction interval.

### 3.1.4 Hoeffding-Bound Test

The Hoeffding Bound proposed by Wassily Hoeffding [28] states that with probability \((1 - \nu)\), the true mean of a random variable of range \(R\) will not differ from the estimated mean after \(N\) independent observations by more than:

\[
\epsilon = \sqrt{\frac{R^2 \ln(\frac{1}{\nu})}{2N}}
\tag{3.5}
\]

This bound offers probabilistic guarantees for the detection of significant changes in the mean of real values (i.e. The difference between the minimum of temporal correlation values recorded between the target stand \(S_i\) and the other stands each 24 hours and the minimum of correlation values in the most recent time interval before the considered prediction interval). The change detection is signaled with probability at most \(\nu\); \(\nu\) is a user-defined parameter. Once a significant change is detected, the subset of stands including their data streams as input to VAR model is automatically updated by replacing the stands which have become irrelevant for the current state with the stands potentially having more influence on the demand at the target stand for the considered...
time prediction interval. Different steps followed in the change detection in the temporal correlation structure are detailed in top-$k$ stands update algorithm.

**Algorithm 1:** Top-$K$ stands update algorithm

**Input:**

$T$: number of periods in the training set

$\theta$: number of time periods contained in a day

**Output:** Top-$k$ stands

1. $min1 \leftarrow \min\{rnormc(X_{i,T}, X_{j_l,T}) = \sqrt{((1 - \text{cor}(X_{i,T}, X_{j_l,T}))/2), l \in \{1, \ldots, k\}}\}$ (Pearson’s Correlation)

2. for each $t \in 1: \theta$ do

   $min2 \leftarrow \min\{rnormc(X_{i,T+t-1}, X_{j_l,T+t-1}), l \in \{1, \ldots, k\}\}$

   $d \leftarrow |min1 - min2|$

   if ($d > \epsilon = \sqrt{\frac{R^2\ln(\frac{1}{\nu})}{2n}}$) (Hoeffding bound) then

   update top-$k$ stands (Using the most recent correlation values)

   $min1 \leftarrow min2$

end

$t++$

Return (top-$k$ stands)

3. end

### 3.2 Best Real-time Induction of Generic and Heterogeneous short-term Taxi-passenger demand Framework

We have already presented four distinct predictive models that can be classified into Time-Varying Poisson models namely, Poisson and Weighted Poisson and Time-Series Analysis models namely, ARIMA and VAR. Using the aforementioned models, we build a **Best Real-time Induction of Generic and Heterogeneous short-term Taxi-passenger demand framework** going by the steps that will be detailed in the next parts.
3.2.1 Model Selection

As it was mentioned before, the use of two non-homogeneous Poisson models may result in a biased ensemble. In addition to that, the performance of time series analysis models depends on the time-changing nature of the demand. For example, when the fluctuations of the demand at a given stand are attributed to the nature of the stand itself in a given time interval, the use of univariate time series model to build the next predictions is more relevant. However this situation may change over time and the fluctuations may become dependent on demand patterns at some other stands which makes more reasonable in this case the use of VAR model. So as a first step, we perform a selection in each time prediction interval, between Poisson and weighted Poisson models and between ARIMA and VAR models, to keep the best performing models. The selection is based on continuously monitoring the residuals of each model. Whenever a significant increase in the residuals of one of the two competing models is detected, this one is discarded. The detection of this drift phenomenon is performed using one of the most standard drift detection algorithms, Page-Hinkley test.

3.2.2 Page-Hinkley Test

The Page-Hinkley test (PH-test) as described by Gama et al. in[24], is a sequential adaptation of the detection of an abrupt change in the average of a Gaussian signal. It allows efficient detection of changes in the normal behavior of a process established by a given model. This test uses the cumulative difference between the observed values and their mean until the instant $T$. Consider $M_1$ and $M_2$ the two competing models and $d = (d_1, \ldots, d_T)$ a vector of the time series variables for the difference between the residuals of $M_1$ and $M_2$ with an aggregation period of $P$ minutes.

$$U_T = \sum_{t=1}^{T} (d_t - \bar{d}_T - \nu)$$  \hspace{1cm} (3.6)

Where $\bar{d}_T$ is the mean of the observations till the moment $T$ and $\nu$ corresponds to the allowed magnitude of changes. The Test monitors the following difference:

$$PH_T = U_T - \min(U_t, t = 1 \ldots T)$$  \hspace{1cm} (3.7)
It signals a change each time the above difference exceeds a given threshold $\lambda$. This detection threshold affects the balance between false and true alarms trigged during this process. Increasing $\lambda$ will entail fewer false alarms, but might miss or delay some changes. Controlling this detection threshold parameter makes it possible to establish a trade-off between the false alarms and the miss detections. Although $\lambda$ is key parameter for making the use of PH-test efficient in change detection problems, in practice it is often difficult to find the optimal $\lambda$ value. The setting of $\lambda$ varies widely depending on the detection problem and the input data.

3.2.3 Sliding-Window Ensemble

The term ensemble is used to identify a set of predictor models (for instance, classifiers, regression models or time series analysis ones) for which individual decisions are in some way combined (typically, by voting or by weighting their outputs) to predict/classify novel time/data points [29].

After selecting the two-best performing models in the previous step, we will introduce them in a well-known sliding-window ensemble, the same that have been used by Moreira et. al in [22].

Consider $M = M_1, M_2, ..., M_z$ to be a set of $z$ models of interest to model a given time series and $F = F_{1,t}, F_{2,t}, ..., F_{z,t}$ to be the set of the predicted values for the next period on the interval $t$ by those models. The ensemble forecast $E_t$ is obtained as

$$E_t = \sum_{i=1}^{z} \left( F_{i,t} \left( 1 - \rho_{i,H} \right) \right), \theta = \sum_{i=1}^{z} (1 - \rho_{i,H})$$

Where $\rho_{i,H}$ is the error of model $M_i$ in the periods contained on the time window $[t - H, t]$ (His a user-defined parameter to define the window size) comparatively to the real service count time series. As the information is arriving continuously for the next periods $t, t + 1, t + 2, ...$ the window will also slide to determine how the models are performing in the last $H$ periods. The contribution of each model to the predictions is weighted according to an evaluation metric of its performance. The Symmetric Mean Percentage Error (sMAPE) is used. This metric will be formally described in Chapter 5 section. The Figure 3.2 summarize the above described steps and illustrates BRIGHT
framework

Figure 3.2: BRIGHT Framework

Conclusion

This chapter sums up the methodology we followed. For each target stand, we select the subset of stands that will include their historical demand as features to perform demand prediction using VAR model using the above explained selection criteria. The
stand selection is updated periodically and also whenever a significant change in the relation between the stands is observed. The change detection is performed basing on Hoeffding Bound test. We build predictions for the demand that will emerge at each stand for the next $P$ minutes using the four predictive models. By monitoring continuously the residuals of each model using Page-Hinkley test which triggers an alarm whenever a significant increase in the residuals of one model comparing the other is detected, we select the best performing model among the time series analysis models and the time-varying Poisson models. The two resulting models are combined later in a sliding window ensemble which enables us to calculate the predicted value of the future emerged services at the target stand in a time horizon of $P$ minutes.
We detailed in the previous chapter the methodology of the present work. The application of this methodology has led to the design of BRIGHT Framework (i.e. Best Real-time Induction of Generic and Heterogeneous short-term Taxi-passenger demand). We tested our framework by selecting two large-size taxi fleets running in two different cities, as case studies.

In case study A, we focused on the data streams of a taxi company running a fleet of 438 vehicles in Porto, Portugal between July 2013 and June 2014. This city is the center of a medium size area (consisting of 1.3 million habitants) which contains 63 taxi stands. In case study B, taxi data stream is collected from a fleet of 1395 taxis running in the second-largest city in Greece, Thessaloníki from Jan. to Nov. 2015. The city holds 68 taxi stands. In this section, we describe the data acquisition process and the preprocessing applied to each case study.

4.1 Case Study A:

4.1.1 Data Acquisition and Preprocessing

The map presented in Figure 4.1 shows the spatial distribution of the 63 taxi stands in the city of Porto, Portugal. A change of the schedule coverage results in one of two
scenarios: (i) a group of days $B$ changes from one coverage to each data chunk arrives with the following six attributes: (1) $TYPE$ – relative to the type of event reported and has four possible values: busy - the driver picked-up a passenger; assign – the dispatch central assigned a service previously demanded; free – the driver dropped-off a passenger and park - the driver parked at a taxi stand. The (2) $STOP$ attribute is an integer with the ID of the related taxi stand. The (3) $TIMESTAMP$ attribute is the date/time in seconds of the event and the (4) $TAXI$ attribute is the driver code; attributes (5) and (6) refer to the $LATITUDE$ and $LONGITUDE$ corresponding to the acquired GPS position.

As preprocessing, a discrete time series of taxi demand services with an aggregation period of $P$ minutes is built. There are three types of accounted events: (1) the busy set directly at a taxi stand; (2) the assign set directly to a taxi parked at a taxi stand and (3) the busy set while a vacant taxi is cruising. We consider both a type 1 and type 2 event as service demanded. However, for each type 2 event, the system receives a busy event a few minutes later – as soon as the driver effectively picked-up the passenger – this is ignored by our system. Type 3 events are ignored unless they occur in a radius of $W$ meters from a taxi stand (where $W$ is a user defined parameter). If it does, it is considered as being a type 1 event related with the nearest taxi stand according the
defined criteria. This was done because many regulations prohibit passengers from being picked-up in a predefined radius around a stop (in Porto, a 50m radius is in place).

4.1.2 Data Analysis

In the case of the city of Porto, the existing regulations oblige the drivers not to randomly cruise in the streets looking for passengers; instead, they have to choose a specific taxi stand out of the 63 existing ones in the city and to wait in this stand for the next passenger after the last passenger drop-off. The three main ways to pick up a passenger are as follows:

- A passenger goes to a taxi stand and picks up a taxi. The regulations also oblige the passengers to pick up the first taxi in line (first in, first out).

- A passenger calls the taxi network central and asks for a taxi for a specific time/location. The parked taxis in the stands have priority over the running vacant ones in the central taxi dispatch system.

- A passenger picks directly a vacant taxi in the street while it is going to a taxi stand.

Statistics about this case study are presented. Figure 4.2 presents the sample distribution of the cruise time of the services required. Table 4.1 details the number of taxi services demanded per daily shift and day type. Table 4.2 contains information about all services per taxi/driver. Additionally, we can state that in this case study 77% of services are directly demanded to taxis parked in taxi stands (and only 23% are assigned while they are cruising). The average waiting time (to pick up passengers) of a taxi parked in a taxi stand is approximately 42 min, while the average time for a service is only 11 min and 12 s. Such low ratio of busy/vacant time reflects the current economic crisis in Portugal and the regulators’ inability to reduce the number of taxis in the city. It also highlights the importance of the predictive system that is presented here, where the shortness of services could be mitigated by obtaining services from the competitors.
The data in Tables 4.1 and 4.2 sustain that, despite the regularity in the service (particularly on weekends), there are major differences among the services that are provided by each driver (i.e., a large variance in service number and profit) related to their different levels of mobility intelligence. Figure 4.2 focuses on the length of the services; 75% of them last 15 min or less. These statistics sustain the importance of a smart decision on the stand-choice problem; an accurate sensor to measure the passenger demand can be a major advantage in urban areas where a highly competitive scenario, like the one described here, is in place.

Figure 4.2: Frequency Distribution of Taxi Cruise Time for the entire data set of Porto, Portugal
### Case Study A

#### Data Acquisition and Preprocessing

<table>
<thead>
<tr>
<th>Day type</th>
<th>Total Services Emerged</th>
<th>Averaged Service Demand per Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>00am to 08am</td>
</tr>
<tr>
<td>Workdays</td>
<td>597094</td>
<td>81665</td>
</tr>
<tr>
<td>Weekends</td>
<td>207047</td>
<td>66492</td>
</tr>
<tr>
<td>All Daytypes</td>
<td>804141</td>
<td>148157</td>
</tr>
</tbody>
</table>

**Table 4.1: Taxi Services Volume (Per Daytype/Shift)**

<table>
<thead>
<tr>
<th>Services per Driver</th>
<th>Case Study A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>10750</td>
</tr>
<tr>
<td>Minimum</td>
<td>102</td>
</tr>
<tr>
<td>Mean</td>
<td>3905</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>1557</td>
</tr>
</tbody>
</table>

**Table 4.2: Taxi Services Volume (Per Driver)**

### Case Study B:

#### Data Acquisition and Preprocessing

The map presented in Figure 4.3 shows the spatial distribution of the 68 taxi stands in the city of Thessaloníki, Greece. The time interval between two adjoining records is 30 seconds. Each record contains information about the following attributes: (1) **taxi-ID**: is a unique identifier for each taxi. The attributes (2)/(3) are the **DATE/TIME** of each event in human time format. The attributes (4)/(5) refer to the **LATITUDE** and **LONGITUDE** of each GPS point. The (7) **STATUS** attribute corresponds to the current status of the taxi, where 1 means occupied and 0 means vacant.

The preprocessing of this data is performed in two steps:

1. **Step 1.** A file of raw GPS records is divided into different files by taxi-ID, with each file standing for one taxi. We convert the date/time records into Unix time format (in seconds) and store them under **TIMESTAMP** attribute. Records of each file are organized by **TIMESTAMP**. A change of value in **STATUS** means an event. The value changing from 0 to 1 represents a pick-up event, and the value changing from 1 to 0 represents a drop-off event. Therefore, the number of services performed per each taxi...
Figure 4.3: Taxi-stand spatial distribution in the city of Thessaloníki, Greece

is a collection of pick-up events.

**Step 2.** To determine the number of services dispatched from each taxi stand, we count the pick-up events that occur in a radius of $W_1$ meters from a taxi stand (where $W_1$ is a user defined parameter). By using GPS coordinates, we calculate the distance between the starting point of the event and each taxi stand. The pick-up event is considered to be dispatched from the nearest taxi stand according to the defined criteria. Otherwise, the event is simply ignored by our system.

### 4.2.2 Data Analysis

Table 4.3 and Table 4.4 contain details about the number of taxi services demanded per daily shift and day type and information about services performed per taxi/driver respectively. Like the case of Porto data set, despite the regularity in the service (particularly on weekends), there are major differences among the services that are provided
### Table 4.3: Taxi Services Volume (Per Daytype/Shift)

<table>
<thead>
<tr>
<th>Day type</th>
<th>Total Services Emerged</th>
<th>Averaged Service Demand per Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>00am to 08am</td>
</tr>
<tr>
<td>Workdays</td>
<td>639922</td>
<td>101727</td>
</tr>
<tr>
<td>Weekends</td>
<td>212072</td>
<td>57119</td>
</tr>
<tr>
<td>All Daytypes</td>
<td>851994</td>
<td>158846</td>
</tr>
</tbody>
</table>

Table 4.3: Taxi Services Volume (Per Daytype/Shift)

<table>
<thead>
<tr>
<th>Services per Driver</th>
<th>Case Study A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum</td>
<td>3972</td>
</tr>
<tr>
<td>Minimum</td>
<td>50</td>
</tr>
<tr>
<td>Mean</td>
<td>873</td>
</tr>
<tr>
<td>Std. Dev.</td>
<td>495</td>
</tr>
</tbody>
</table>

Table 4.4: Taxi Services Volume (Per Driver)

by each driver (i.e., a large variance in service number and profit) related to their different levels of mobility intelligence. The presented statistics shows the utility of our framework in enhancing the profitability of taxi drivers by providing the same information about the spatiotemporal distribution of passengers to all the drivers at the same time.

### Conclusion

In this section, we described the acquisition and the preprocessing steps of the data sets of two large-size taxi fleets running in the cities of Porto, Portugal and Thessaloníki, Greece. In the next chapter, we will present the results of the evaluation of our framework on both data sets.
Experimental Results

Introduction

In this chapter, we will thoroughly detail the different steps followed to validate our framework. We will start by presenting the hardware and the software environments in which we conducted our experiments, as well the experimental set-up. We will present and discuss the results we achieved to demonstrate the efficiency and the usefulness of our framework.

5.1 Experiments

5.1.1 Hardware Environment

Our experiments were conducted using a desktop machine with the following characteristics.

- CPU Intel Core i3 3.2 GHz
- 8 GB RAM
- Hard Drive 800 GB
- System Type 64 bit
5.1.2 Software Environment

The experiments were conducted using the R language [30]. R is a programming language and environment for statistical computing and graphics. It has been lifted to be the single most important tool for computational statistics, visualization and data science. R is available as Free Software under the terms of the Free Software Foundation’s GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems, Windows and MacOS.

Thanks to the fact that it is a programmable environment using command-line scripting, we can store a series of complex data-analysis steps in R. During the last decade, it has become the most popular language for data science and an essential tool for Finance and analytics driven at the most popular companies such as Google, Facebook, LinkedIn and Amazon.

One of R’s strengths is the fact that it is highly extensible and offers a wide range of facilities for data manipulation, computations and graphical display. What is more, it includes an effective data handling and storage facility, operators for calculations on frames, arrays, in particular matrices, a large collection of tools for data analysis. In short, it is a well-developed, simple and effective programming language, which includes conditionals, loops and more. We used R for the implementation of the statistical approach because R provides us with ready-made packages that contain traditional and modern statistical models (regression, ARIMA...). Researches all around the world are working to improve R-packages by developing the latest methods in statistics and predictive modeling. To date, R has numerous packages used in predictive modeling. As far as this part of the project is concerned, we used the following packages:

1. “stats” package: is a built-in R package that provides the user with all the time series models and all the required statistical tests.

2. “tseries” package [32]: is used in order to exploit the “adf.test” function that helps to test the stationarity of our data.

3. “forecast” package [31]: provides forecasting functions and linear models for
analyzing univariate time series forecasts including exponential smoothing and ARIMA modeling.

4. **“MTS” package [33]**: Multivariate Time Series (MTS) is a general package for analyzing multivariate linear time series and estimating multivariate volatility models. For the multivariate linear time series analysis, the package performs model specification, estimation, model checking, and prediction for many widely used models, including vector AR models, vector MA models, vector ARMA models, Seasonal Vector ARMA models, VAR models.

5. **“ggplot2” package [34]**: is a plotting system in R. It takes care of many of the important details, like drawing legends and axis’ names, as well as providing a powerful graphic tools.

6. **“ggmap” package [35]**: A collection of functions to visualize spatial data and models on top of static maps from various online sources (e.g Google Maps and Stamen Maps). It includes tools common to those tasks, including functions for Geo-location and routing. We have used this package to visualize GPS data coordinates.

### 5.1.3 Experimental Setup

We used an $H$-sized sliding window to measure the error of our model before each new prediction about the service count on the next period (the metrics used to do so are defined in the section 5.2). Each new real count was used to update our predicting model. The predictions are produced for the last four months for each case study. An aggregation period of 30 minutes was set (i.e. a new prediction of the event count is produced each 30 minutes; $P=30$) and a radius of 100m ($W=W_1=100$) was used to perform the preprocessing step for both case studies.

The time-varying Poisson averaged models (both weighted and non-weighted) were updated every 24 hours. The $\alpha$ parameter of the Weighted Poisson model is defined for each stand using an hyper-parameter tuning process which employs gird search over 100 distinct samples as admissible values for $\alpha \in [0, 1]$ to determine the value that
maximizes our predictive performance on each stand using a validation set (i.e. dataset containing data very similar to the one tested on our experiments). The $\gamma$ value was set respecting the following definition:

$$\gamma = \max(N): \omega \geq 0.01$$ (5.1)

The ARIMA model $(p; d; q$ values and seasonality) was firstly set and then updated each $24h$ by learning/detecting the underlying model (i.e. autocorrelation and partial autocorrelation analysis) running on the historical time series curve of each stand during the last two weeks (i.e. period $t - 2\theta, t$). To do so, we used an automatic time series function in the forecast package - auto-arima – with the default parameters. The weights/parameters for each model are specifically fit for each period/prediction using the function arima from the package stats.

The same approach was followed for the set-up of VAR model. Before specifying the order $p$ and the weights, we define for each stand the set of that include their historical time series of the demand to build VAR model, using the two previously defined selection criteria the correlation threshold and the top-$k$ number of stands. Those two selection criteria are set for each stand using a grid search procedure to choose ideal values of the two parameters that maximizes the predictive performance of VAR model. The grid search process is performed for each month of the test set using the previous month as a validation set to evaluate the performance of VAR model with a fixed grid of those parameters. We investigate possible correlation threshold values $\in [0.1, 0.9]$ with a step of 0.05 and top-$k$ values in $[1, 10]$. After fixing those parameters, the subset of stands selection is performed each 24 hours. The order $p$ and the set of matrices weights $\Pi$ are initially set and updated each 24 hours by building the underlying model using the cross-correlation functions analysis applied to the set of historical time series of the demand at the top-$k$ stands for the last two weeks (i.e. period $t - 2\theta, t$). We used ccf function from the built-in R package [stats] for the $p$ order selection and VAR and VARpred in the MTS package for the model and the predictions building. The confidence interval of the Hoeffding Bound test used to monitor significant temporal correlation changes in each time prediction interval is set to 5%.

The threshold $\lambda$ and the allowed magnitude of changes $\nu$ of PH-test are set fol-
Table 5.1: Description of the Learning Periods

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sliding Window</th>
<th>Nr. of Periods Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson Mean</td>
<td>ALL Data 1, t</td>
<td>N/A: It is calculated incrementally</td>
</tr>
<tr>
<td>W.Poisson Mean</td>
<td>Last 8 weeks</td>
<td>γ = 8</td>
</tr>
<tr>
<td>ARIMA</td>
<td>Last two weeks</td>
<td>2 * θ</td>
</tr>
<tr>
<td>VAR</td>
<td>Last two weeks</td>
<td>2 * θ</td>
</tr>
<tr>
<td>Ensemble</td>
<td>Last four hours</td>
<td>H = 8</td>
</tr>
</tbody>
</table>

lowing traffic expert’s suggestions. Finally, a sliding window of 4 hours (H = 8) is considered in the ensemble.

Table 5.1 summarizes the information about the learning periods that are used by each algorithm.

5.2 Evaluation Metrics

Well-known error measurements were employed to evaluate our output. Consider $R = R_{k,1}, R_{k,2}, \ldots, R_{k,t}$ to be a discrete time series (aggregation period of $P$-minutes) with the number of services predicted for a taxi stand of interest $k$ in the period $X = X_{k,1}, X_{k,2}, \ldots, X_{k,t}$ the number of services actually emerged in the same conditions.

- **The Root Mean Square Error (RMSE)** is defined for each stand $k$ as:

  $$RMSE_k = \sqrt{\frac{\sum_{i=1}^{t} (R_{k,i} - X_{k,i})^2}{t}} \quad (5.2)$$

- **Normalized root mean square error (NRMSE)** can be defined using the above formulation of the error:

  $$NRMSE_k = \frac{RMSE_k}{\bar{X}} \quad (5.3)$$

- **Symmetric Mean Percentage Error (sMAPE)** is defined for each stand $k$ as follows:

  $$sMAPE_k = \frac{1}{t} \sum_{i=1}^{t} \frac{|R_{k,i} - X_{k,i}|}{R_{k,i} + X_{k,i}} \quad (5.4)$$

We use the re-defined formulation of the original sMAPE measure that was introduced in [22]. In fact, this metric can be intolerant to small magnitude errors.
(e.g. if two services are predicted on a given period for a taxi stand of interest but no one emerges, the error measured during that period would be 1). To produce more accurate error measures about series containing very small numbers (i.e. stands with low demand), a Laplace estimator [36] is frequently added to eq 5.4. In our case, we perform such normalization by adding a constant \( c \) to the denominator (i.e.: originally, it was added to the numerator to estimate a success rate [36]. The \( sMAPE_k \) (i.e.: the error measured on the time series of services predicted to the stand \( k \)) can be defined as:

\[
sMAPE_k = \frac{1}{t} \sum_{i=1}^{t} \frac{|R_{k,i} - X_{k,i}|}{R_{k,i} + X_{k,i} + c} \tag{5.5}
\]

where \( c \) is a user-defined constant. To simplify the theorem application, we consider its most common use: \( c = 1 \) [36] and it is the same value used in [22].

The above metrics are focused just on one time series for a given taxi stand \( k \). However, the results presented below use an averaged error measure based on all stands series – \( AG \). Consider to be an error metric of interest. \( AG_{\beta,t} \) is an aggregated metric given by a weighted average of the error measured in all stands in the period \( 1,t \). It is formally presented in the following equations:

\[
AG_{\beta,t} = \sum_{i=1}^{N} \frac{\beta_{t,k} \cdot \psi_k}{\Psi} \tag{5.6}
\]

\[
\psi_k = \sum_{i=1}^{t} X_{k,i}, \Psi = \sum_{k=1}^{N} \psi_k \tag{5.7}
\]

where \( \psi_k \) is the total of services emerged at the taxi stand \( k \); \( \beta_{t,k} \) is the error measured by \( \beta \) at the stand \( k \) and \( \Psi \) is the total of services emerged at all stands so far.

### 5.3 Results and Discussion

This section thoroughly presents the results we achieved which are illustrated by the following figures and tables. In the discussion subsection, we will explain the results separately to clarify every finding.
5.3.1 Results

The results are presented over five distinct perspectives:

1. Stands selection process and the performance of the Hoeffding Bound test with a concrete example

2. An example of drift detection in the residuals of two predictive models using the Page-Hinkley (PH) test

3. The averaged error per methods and per error measure for each case study

4. A comparative analysis of our framework performance versus the state-of-art sliding-window ensemble presented by Moreia-Matias et. al in [22]

5. A direct analysis of some output examples

First, Figure 5.1 shows the stand selection process for the stand 11 in Porto on a typical work day and the introduced changes on the initially selected subset of stands once significant changes in the correlation between the stands are detected (i.e. Hoeffding Bound test triggers an alarm).

Second, considering the predicted number of services generated by the base-line predictive models, we illustrate in Figure 5.2(a) and 5.2(b) the performance of the Page-Hinkley test on a typical work day in the stand 36 in Porto and the stand 1 in Thessaloníki, respectively. In Figure 5.2(a), Page-Hinkley(PH)-test triggers an alarm whenever VAR residuals exceed ARIMA residuals. In Figure 5.2(b), Page-Hinkley(PH)-test triggers an alarm whenever Poisson model residuals exceed Weighted Poisson model residuals.

Third, the error measured for each model is presented in Table 5.2. The results are firstly presented globally and then per day shift. The results were aggregated using the $AG_β$ previously defined.

Fourth, the Figure 5.3 presents a comparison between our model and the base-line models. The Figure 5.4 shows a comparison between our model and the S.o.A. ensemble [22] on a typical workday. The presented values were calculated using the same
<table>
<thead>
<tr>
<th>Case Study</th>
<th>Model</th>
<th>Evaluation Metrics</th>
<th>Periods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>24h</td>
</tr>
<tr>
<td>Case Study A</td>
<td>Poisson Mean</td>
<td>24.05%</td>
<td>21.84%</td>
</tr>
<tr>
<td></td>
<td>W.Poisson Mean</td>
<td>24.32%</td>
<td>21.83%</td>
</tr>
<tr>
<td></td>
<td>ARIMA</td>
<td>27.31%</td>
<td>27.62%</td>
</tr>
<tr>
<td></td>
<td>VAR</td>
<td>25.53%</td>
<td>22.02%</td>
</tr>
<tr>
<td></td>
<td>S.o.A Ensemble</td>
<td>23.82%</td>
<td>21.70%</td>
</tr>
<tr>
<td></td>
<td>BRIGHT</td>
<td>22.91%</td>
<td>19.91%</td>
</tr>
<tr>
<td>Case Study A</td>
<td>Poisson Mean</td>
<td>11.77%</td>
<td>11.48%</td>
</tr>
<tr>
<td></td>
<td>W.Poisson Mean</td>
<td>12.20%</td>
<td>11.87%</td>
</tr>
<tr>
<td></td>
<td>ARIMA</td>
<td>12.41%</td>
<td>12.31%</td>
</tr>
<tr>
<td></td>
<td>VAR</td>
<td>13.46%</td>
<td>13.17%</td>
</tr>
<tr>
<td></td>
<td>S.o.A Ensemble</td>
<td>11.46%</td>
<td>11.02%</td>
</tr>
<tr>
<td></td>
<td>BRIGHT</td>
<td>10.76%</td>
<td>8.73%</td>
</tr>
<tr>
<td>Case Study A</td>
<td>Poisson Mean</td>
<td>24.99%</td>
<td>23.48%</td>
</tr>
<tr>
<td></td>
<td>W.Poisson Mean</td>
<td>27.13%</td>
<td>25.53%</td>
</tr>
<tr>
<td></td>
<td>ARIMA</td>
<td>26.93%</td>
<td>28.58%</td>
</tr>
<tr>
<td></td>
<td>VAR</td>
<td>26.12%</td>
<td>25.04%</td>
</tr>
<tr>
<td></td>
<td>S.o.A Ensemble</td>
<td>24.68%</td>
<td>23.72%</td>
</tr>
<tr>
<td></td>
<td>BRIGHT</td>
<td>24.09%</td>
<td>23.16%</td>
</tr>
<tr>
<td>Case Study B</td>
<td>Poisson Mean</td>
<td>5.45%</td>
<td>7.40%</td>
</tr>
<tr>
<td></td>
<td>W.Poisson Mean</td>
<td>5.94%</td>
<td>7.92%</td>
</tr>
<tr>
<td></td>
<td>ARIMA</td>
<td>5.39%</td>
<td>7.51%</td>
</tr>
<tr>
<td></td>
<td>VAR</td>
<td>5.74%</td>
<td>7.14%</td>
</tr>
<tr>
<td></td>
<td>S.o.A Ensemble</td>
<td>5.25%</td>
<td>7.08%</td>
</tr>
<tr>
<td></td>
<td>BRIGHT</td>
<td>5.02%</td>
<td>6.93%</td>
</tr>
</tbody>
</table>

Table 5.2: Error Measured on The Models Using sMAPE and NRMSE
Figure 5.1: Illustration of the stand selection process for 24h period for the stand 11, Porto

4-h sliding window of the ensemble (the error of instant \( t \) is the error that is measured at period \([t - H,t]\), \( H = 8 \)).

Finally, examples of distinct weekly analysis of the discrepancies between the predicted demand and the services actually emerged for both case studies are displayed in Figure 5.5.

### 5.3.2 Discussion

Figure 5.1 enables us to visualize the Hoefding Bound test effect on the stand selection process. The stand selection is performed each 24 hours for each target stand. Like the studied example shows, the Hoefding Bound test triggers an alarm 3 times (i.e. periods 17, 24 and 36) during 24 hour and we can notice effective changes in the subset of the selected stands which shows the effectiveness of our framework in updating
the input of VAR model and as a result its specification and its estimation whenever a significant change in the relation between the stands is observed with each prediction period.

Figure 5.2 shows the reactivity of the Page-Hinkley test to drifts in the performance of the predictive models. It also shows a small false triggered alarm rate. This illustrates the efficiency of the use of Page-Hinkley test in drift detection problems as well as, the efficiency of our parameters setting in each case study.

BRIGHT is always the best model in every day shift and in the whole studied period. For both case studies, BRIGHT is always the model with the least error independently of the used error measure metric. The overall performance is very good. A error analysis per stand shows a maximum value of the error of 28.83% in case study A and a value of 27.44% in case study B. The models just present slight discrepancies within the daily shifts. BRIGHT methodology is robust compared to the remaining models.

In Figure 5.3, it is possible to identify multiple points where BRIGHT maintained its robustness while the other methods presented a significant increase in the prediction error. In Figure 5.4, it is possible to see that the performance of BRIGHT is better than the S.o.A. Ensemble in almost all the points.

Figure 5.5 shows two distinct scenarios to compare the forecasted and the real demand. In Figure 5.5(a), the demand corresponds to a regular taxi stand where services have a usual pattern (low in the beginning of the first day shift and the end of the third day shift and high with some peaks in the second day shift); in Figure 5.5(b), the chart corresponds to a completely irregular stand behavior (changes in the demand pattern in the weekend). The two examples illustrate that our framework can correctly forecast the demand in distinct scenarios, periods, weeks.

In this work, the output variable is the number of services that will emerge in a specific taxi stand in a time-horizon of $P$ minutes. This variable was chosen due to the stand relevance in both case studies (where the majority of services is directly dispatched from taxi stands). However, our framework is valid in the case where the services are demanded from different areas in the city since the mathematical model does not depend on how the service historical spatial distribution (i.e., by stand or by
spatial cluster) but only on the demand time series with an aggregation period of $P$ minutes (which is user defined parameter).

**Conclusion**

Along this chapter, we have defined our implementation of the framework. We presented the case study as well as the experiments we conducted. Next, we put forth the results we achieved. These results are relevant to different steps of the framework and demonstrate its validity through different perspectives.

The model was tested on GPS data collected from taxi companies operating in Porto, Portugal and Thessaloniki, Greece. BRIGHT model has presented a more than satisfactory performance, correctly predicting 268750 and 285343 services in the test period in Porto and Thessaloniki, repetitively. The aggregated error measured in both cases is lower than 25.5%. The model has a better performance than the S.o.A Ensemble. The aggregated error was reduced in both case studies.

This work presents a novel contribution to the existing state of the art framework presented in [22] and reduces its limitations. This has been achieved due to:

1. Exploring dependencies between different taxi stands over time;

2. The introduction of a multivariate time series analysis model, namely Vector Autoregressive VAR; Updating its formulation to the context of taxi services prediction on specific taxi stands;

3. The use of drift detection methods, namely the Hoeffding Bound test and the Page-Hinkley algorithm to monitor the performance of the predictive models and to respond the time-changing environment of the predictions;

4. The building of a two-stage ensemble framework using the same ensemble model formulation to reduce bias in the S.o.A Ensemble and to ameliorate its performance;
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.2: Illustration of Performance of the Page-Hinkley (PH)-test

(a) Performance of the Page-Hinkley (PH)-test on ARIMA and VAR models (Stand 36 Porto)

(b) Performance of the Page-Hinkley (PH)-test on Poisson and Weighted Poisson models (Stand 1 Thessaloniki)
CHAPTER 5. EXPERIMENTAL RESULTS

Figure 5.3: BRIGHT evaluation on a typical work day (Porto)

Figure 5.4: Comparison between BRIGHT and S.o.A. Ensemble on a typical work day (Porto)
Figure 5.5: Weekly comparison between the forecasted services and the real-emerged services on two distinct scenarios/taxi stands and weeks
This work presents an adaptive Learning approach for Short-term Taxi-passenger Demand Prediction. The output of the designed framework is the number of services that will emerge in each specific taxi stand for a short time horizon of $P$ minutes. Our main contributions compared to the existing state of the art are the identification and the use of dependencies between different taxi stands over time and the introduction of a multivariate time series analysis model, namely **The Vector Autoregressive VAR**, in addition to **drift detection methods**, in order to build accurate predictions in a **real-time** context. Our framework was tested on real-world data sets. The obtained results were both efficient and successful compared to the existing S.o.A framework. Our project led to filing an international patent “**Method to Control Electrical Vehicle Fleets to deliver Inexpensive On-Demand Transportation Services**” at the United States Patent and Trademark Office and we are currently preparing for the submission of a paper.

In spite of the promising results obtained on both case studies, there are still some issues to handle on future research. Both Porto and Thessaloniki are an interesting case studies. However, both are a mid-sized cities. Our framework should be tested on another type of case studies with a larger volume of services and where the demand exceeds the supply. Furthermore, this framework depends on a comprehensive set of parameters. Some of them suffered a tuning stage before using. It is important to develop automatic frameworks to overcome such limitations in the near future. Such topics are still open research questions.
References


[2] https://www.otexts.org/fpp/8/1


REFERENCES


[26] Cryer, J., Chan, K.,


[33] https://cran.r-project.org/web/packages/MTS/MTS.pdf

[34] https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
