TÉCNICO LISBOA

Clustering of Load Curves to Support Demand and Generation Forecast

Ana Catarina Constantino Vaz

Thesis to obtain the Master of Science Degree in

Mathematics and Applications

Supervisor(s): Prof. Maria da Conceição Esperança Amado
Dra. Margarida de Almeida Pedro

Examination Committee

Chairperson:	Prof.António Manuel Pacheco Pires
Supervisor:	Prof. Maria da Conceição Esperança Amado
Member of the Committee:	Prof. Cláudia Rita Ribeiro Coelho Nunes Philippart
	Marco André Gonçalves Pinheiro

November 2017

Acknowledgments

I want to express my gratitude to:

My parents and brother, for their unconditional and constant support through my Bachelor and Master at Técnico.

Professor Conceição Amado, for her availability during the whole period of elaboration of this thesis, for her valuable suggestions and her kindness.

Margarida Pedro, from EDP, for the provided data and availability to clarify all doubts that occurred.

Last but not least, I want to thank my Bachelor and Master colleagues, for the companionship demonstrated everyday.

Resumo

Cada cliente de uma empresa de eletricidade exige uma certa quantidade de eletricidade em cada instante. Portanto, é necessário gerar energia de forma a satisfazer a procura. Com o intuito de otimizar os recursos que permitem a geração de energia, torna-se importante conhecer o valor da procura em avanço. Assim, este estudo tem como objetivo o ajuste de modelos que explicam o consumo de eletricidade dos clientes e que permitem a previsão do mesmo um dia adiante.

Porém, perante uma grande carteira de clientes, torna-se pouco prático analisar cada consumo individualmente. Como tal, recorremos a métodos de clustering para agrupar clientes com base na semelhança de consumo e, para cada grupo, propomos um modelo representativo. Dois tipos de modelos serão comparados com base na precisão das previsões: modelos aditivos generalizados e modelos auto-regressivos integrados de médias móveis.

Palavras-chave: Previsão do Consumo de Eletricidade, Séries Temporais, Agrupamento Hierárquico, Modelos Aditivos Generalizados.

Abstract

Each customer of an electricity company demands a certain amount of electricity at every instant. So, this amount needs to be generated in order to suit the customer's demands. In order to optimize the use of energy generation resources, it is necessary to know the demanded quantity in advance. Hence, this study is aimed at fitting models that explain customer's electricity consumption and that enable the forecast of one day ahead consumption.

However, for a large customer portfolio, it becomes impractical to analyse each consumption individually. Therefore, we make use of clustering methods to group customers by similarity of consumption and, for each cluster, we propose a representative model. Two types of models will be compared based on their forecast accuracy: generalized additive models and autoregressive integrated moving average models.

Keywords: Load Forecasting, Time Series, Hierarchical Clustering, Generalized Additive Models.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This Chapter is divided in three parts. Firstly, Section 1.1 is intended to explain the problem that motivates the study of this thesis. Several authors have faced the same or similar problems until today, therefore it is important to understand the level of development of ideas and methods used for the achievement of a solution. The description of this level of development is known as state-of-the-art and it is detailed in Section 1.2. Our aim is to present a method that outperforms the ones presented in the state-of-the-art. So, finally, the last Section of this Chapter (Section 1.3) contains the description of how the thesis is structured.

1.1 Motivation

Each customer of an electricity company demands a certain amount of electricity at every instant, therefore this amount needs to be generated in order to suit the customer's demands. In large scale and when the generation is greater than the demand, there are still major technical challenges that make it impractical to store the excess of electricity. So, in order to avoid losses and to optimize the use of energy generation resources, it is important to reduce the mismatch between procured generation and demand. Having this in mind, given data that represents one customer's electricity consumption from previous years until today, our aim is to build a statistical model that best explains the data and to forecast future values of electricity consumption accurately. One of the major difficulties in attaining a good model accuracy is related with the unpredictability of consumption, especially during weekends, public holidays or periods of holidays, for instance, during summer.

On one hand, we will study models that only consider historical data of electricity consumption. On the other hand, we will study models that include as well external variables. In this latter case, understanding the underlying variables that motivate different electricity consumption behaviours might be essential in forecasting consumption. In particular, we will understand the influence of weather variables in electricity consumption, in order to introduce these in the model.

Now consider that we have millions of customer's consumption to forecast. It becomes impractical to analyse the demand of each customer individually. Finding groups of customers with similar consump-

tion behaviours allows us to reduce the number of behaviours that need to be analysed. This grouping procedure, known as clustering, enable us to create only one statistical model for each group. In order to do the clustering, it is necessary to state what do we define by similarity of consumption behaviours and which are the best techniques that perform the grouping.

Finally, having obtained the right customer's partition allied with the best model that explains each cluster's data, we are able to foresee consumer behaviours in the near future.

1.2 State-of-the-Art

Load forecasting of a large set of different customers requires the realization of two main steps. The first step consists in clustering customer load profiles, whereas the second one consists in fitting a model that explains the consumption of each cluster and forecasting short term electricity load.

Clustering is an unsupervised learning task aimed at the partition of a set of unlabelled data objects into homogeneous groups or clusters. This partition is performed in such a way that objects in the same cluster are more similar to each other than objects in different clusters, according to some defined criterion.

Hyndman et al. [2006] proposed a method for clustering time series based on their structural characteristics. This method extracts global measures of the time series (such as trend, seasonality, periodicity, serial correlation, skewness, kurtosis, chaos, non-linearity, and self-similarity), thus reducing the dimensionality. This way, it is much less sensitive to missing or noisy data. The feature measures are obtained from each individual series and can be fed into arbitrary clustering algorithms, including an unsupervised Neural Network algorithm or Hierarchical Clustering algorithm.

Rasanen et al. [2010] presented a clustering method based on the combination of Self-Organizing Maps (SOM) with two different clustering methods: k-means and hierarchical clustering with the Euclidean distance. The SOM method was considered as a suitable intermediate step before the clustering process, since it is an algorithm characterized by robustness (noise reduction) and computational efficiency. The proposed methodology was applied on a dataset consisting of hourly measured electricity use data, for 3,989 small customers, during 1 year.

Goude and Cugliari [2016] proposed a two stage strategy. In the first stage it was created a large number of super customers using the Partitioning Around Medoids (PAM) algorithm. In the second stage, hierarchical clustering was applied to these super customers with the wavelet coherence distance, where similarities between curves were measured with a functional distance using discrete wavelet transform for multiresolution approximation. Wavelets compress electricity curves efficiently while preserving information useful for clustering. The strategy was applied on a dataset consisting of half hourly measured electricity use data, for 25,011 business customers, during 2 years and 6 months.

While the two former paragraphs mention clustering methods that were applied on raw electricity data, Chicco [2012] proposed a pre-clustering phase, characterized by setting up a normalised Representative Load Pattern (RLP), in order to get load patterns comparable in terms of their shape. The RLP was computed by dividing the typical daily load pattern by its reference power, where the reference

power is defined as the peak value of the typical daily load pattern. The Hierarchical Clustering was then applied to these RLP's.

Notice that most of the clustering task requires iterative procedures to find locally or globally optimal solutions from a high-dimensional data sets. Also, it may be necessary many experimentations with different algorithms or with different features of the same data set. Hence, clustering is computationally expensive and saving computational complexity is a significant issue for the clustering algorithms. Therefore, the parallelization of clustering algorithms is a very practical approach. Kim [2009] proposed parallel versions of the Hierarchical Clustering, the k-means and also for the Neural Networks.

For the modelling and forecasting short term electricity load, the second main step mentioned in the first paragraph of this Section, Huang and Shih [2003] proposed an ARMA model, whereas Taylor [2012] proposed the use of an exponential smoothing formulation. These two modelling approaches use only historical load data, so they are called univariate models. The following approaches use models that explain electricity consumption based on different variables. Bianco et al. [2009] proposed linear regression models to forecast electricity consumption in Italy, using economic and demographic variables. The variables of the model were historical electricity consumption, Gross Domestic Product (GDP), GDP per capita, and population. On the other hand, Yannig Goude and Sinn [2012] proposed a weather-based Generalized Additive Models (GAM) and evaluated the methodology on 5 years of electricity load data. The results showed that GAM outperforms the other state-of-the-art methods in terms of model tracking and prediction accuracy.

There is a continuing effort to improve the accuracy of the consumption forecasts. In this study, we begin by normalizing the load curves in order to do the clustering, based on Chicco [2012], with a modified formula in the way of normalizing though, and after that, we present a GAM for each cluster, based on Yannig Goude and Sinn [2012].

1.3 Structure of the Thesis

This dissertation is structured in the following way. Chapter 2 describes the methodology used in this thesis to model and forecast a large set of load curves. Chapter 3 presents the results, supported by several tables and graphics, of clustering 25 different load curves, followed by fitting a model that explains the consumption of each cluster and forecasting one-day ahead consumption. The concluding remarks are stated in Chapter 4.

Chapter 2

Methodology for Modelling and Forecasting a Large Set of Time Series

A time series is an ordered sequence of observations (Wei, 2006). Time series occur in a variety of fields, for instance, in finance we can observe stock prices, whereas in the medical field we can observe electrocardiogram tracings. In particular, during this study, we encounter time series in the electrical and also meteorological fields. In Section 2.1, formal definitions of time series and related concepts are presented.

Our objective is to identify a model that best fits observations of a given time series and to predict its future values. Having this is mind, we compare the performance of two kinds of models. On one hand, we only consider historical data by fitting time series models, which are explained in Sections 2.2 and 2.3. On the other hand, in Section 2.4, we consider generalized additive models, which allow the existence of external variables.

Since we will be dealing with a large set of time series, it is impractical to identify models for each one of them. This requires that we group time series according to their similarity, thus, a clustering method must be used. In the last Section of this Chapter, it is described clustering techniques used in time series (Section 2.5).

2.1 Fundamental Concepts in Time Series

In order to properly define time series, we begin by defining stochastic processes, according to Žitković (2010).

Definition 2.1.1. Let $\mathcal{T}$ be a subset of $[0, \infty)$. A family of random variables $\{X_t : t \in \mathcal{T}\}$, indexed by $\mathcal{T}$, is called a stochastic (or random) process. When $\mathcal{T} = \mathbb{Z}$, $\{X_t : t \in \mathcal{T}\}$ is said to be a discrete-time process, and when $\mathcal{T} = [0, \infty)$, it is called a continuous-time process.

Thus, a time series is a sample function, or realization, from a certain stochastic process. In this study, we will deal with discrete-time processes. According to Brockwell and Davis (2002), a time series $\{X_t : t \in \mathbb{Z}\}$ may be decomposed as:

$$X_t = m_t + s_t + \epsilon_t, t \in \mathbb{Z} \tag{2.1}$$

which is called the classical decomposition model, and where m_t is a slowly changing function known as a trend component, s_t is a function with known period referred to as a seasonal component, and ϵ_t is a random noise component.

In general, for $h \in \mathbb{Z}$, ϵ_t and ϵ_{t+h} are dependent, since, in practice, future events are influenced by previous events. So, it is necessary to introduce the concept of stationary. Loosely speaking, we say that a time series is stationary if it has statistical properties similar to those of the time-shifted series.

Definition 2.1.2. A time series $\{X_t\}$ is strictly stationary if, for every $n \geq 1$ and $h, t_1, ..., t_n \in \mathbb{Z}$, $(X_{t_1}, ..., X_{t_n})$ and $(X_{t_{1+h}}, ..., X_{t_{n+h}})$ have the same joint distribution.

This latter definition introduces a strong sense of stationarity which is described in terms of a distribution function. An example of a strictly stationary process is a sequence of i.i.d. random variables, which usually does not exist in time series. Other than the i.i.d. case, it is very difficult to verify a joint distribution function from an observed time series. Thus, in time series analysis, we use a weaker sense of stationarity in terms of the moments of the process. Considering only the first and the second moments of the time series, we have the definition of weakly stationarity.

Definition 2.1.3. A time series $\{X_t\}$ is weakly stationary if:

1. the expected value μ_t is finite and constant, and

2. $Cov(X_t, X_{t+h})$ does not depend on t for each h.

Whenever we use the term stationary we shall mean weakly stationary. This concept leads us to the definition of autocovariance function (ACVF) of $\{X_t\}$ at lag h, which is given by, when $Var(X_t) < \infty$:

$$\gamma_h = Cov(X_t, X_{t+h}) = E[(X_t - E[X_t])(X_{t+h} - E[X_{t+h}])], h \in \mathbb{Z}. \tag{2.2}$$

In addition, since ACVF is sensitive to the units in which the observations are measured, we define the autocorrelation function (ACF) of $\{X_t\}$ at lag h as:

$$\rho_h = \frac{\gamma_h}{\gamma_0} = \frac{Cov(X_t, X_{t+h})}{Var(X_t)}, h \in \mathbb{Z}. \tag{2.3}$$

The notion of white noise is important in the analysis of time series and makes use of these concepts.

Definition 2.1.4. A time series $X = \{X_t, t \in \mathbb{Z}\}$ is said to be white noise with expected mean μ and

variance σ^2, and we write $X \sim WN(\mu, \sigma^2)$, if $E[X_t] = \mu$ and, for every $h \in \mathbb{Z}$,

$$\gamma_h = \begin{cases} \sigma^2 & , h = 0 \\ 0 & , h \neq 0 \end{cases}$$

which means that the observations taken over time are uncorrelated. Note also that a white noise time series is stationary, since it satisfies both conditions of definition 2.1.3.

Besides ACVF and ACF, there is also the concept of partial autocorrelation function (PACF), which measures the effects of the relationship between X_t and X_{t-h}, when the effects of the lags $1, 2, 3, ..., h-1$ are removed (Hyndman and Athanasopoulos, 2014). The partial autocorrelation function of a stationary time series $\{X_t\}$ is the function α defined by the equations

$$\alpha(0) = 0 \tag{2.4}$$

$$\alpha(h) = \phi_{hh}, h \geq 1, \tag{2.5}$$

where ϕ_{hh} is the hth partial autocorrelation coefficient of the linear regression model:

$$X_{t+h} = \phi_{h1} X_{t+h-1} + ... + \phi_{hh} X_t + \epsilon_{t+h}, \tag{2.6}$$

with $\{\epsilon_t, t \in \mathbb{Z}\}$ independent and identically distributed to a $Normal(0, \sigma_\epsilon^2)$ and ϵ_{t+h} independent of $\{X_{t+h-j} : j \geq 1\}$.

The measures ACVF, ACF and PACF are very important for characterizing time series. In practical problems, we start with observed data $\{x_1, ..., x_n\}$, for which we want to select a model that reflects the degree of dependence in the data. One of the important tools we use is the sample autocorrelation function (sample ACF) of the data. If we believe that the data are realized values of a stationary time series $\{X_t\}$, then the sample ACF will provide us with an estimate of the ACF of $\{X_t\}$ (Brockwell and Davis, 2002). The sample mean, the sample autocovariance function and the autocorrelation function are given, respectively, by Equations 2.7, 2.8 and 2.9, respectively.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{2.7}$$

$$\hat{\gamma_h} = \frac{1}{n} \sum_{i=1}^{n-h} (x_t - \bar{x})(x_{t+h} - \bar{x}), h = 0, 1, ... \tag{2.8}$$

$$\hat{\rho_h} = \frac{\hat{\gamma_h}}{\hat{\gamma_0}}, h = 0, 1, ... \tag{2.9}$$

and, finally, the sample partial autocorrelation function can be obtained from the sample autocorrelation function, using the method of Dublin-Levinson (see Pacheco, 2001).

2.2 Time Series Models

This Section is devoted to identify models that best fit a given stationary or non-stationary time series.

2.2.1 Models for Stationary Time Series

As described in Brockwell and Davis [2002], the key role in time series analysis is played by processes whose properties, or some of them, do not vary with time. So, given a series that typically is not deterministic but contains a random component, and if this random component is stationary, then we can develop powerful techniques to forecast its future values. This Subsection aims at presenting models that fit stationary time series.

Given a time series $\{X_t\}$, one first approach is to estimate and extract the deterministic components m_t and s_t of the classical decomposition in the hope that the residual or noise component ϵ_t will turn out to be a stationary time series. If this is not the case, then an alternative approach, developed by Box and Jenkins, consists in applying differencing operators repeatedly to the series $\{X_t\}$ until the differenced observations resemble a realization of some stationary time series. The following paragraphs are aimed in detail the latter approach.

We define the lag-1 difference operator ∇ by

$$\nabla X_t = X_t - X_{t-1} = (1 - B)X_t, \tag{2.10}$$

where B is the backward shift operator, $BX_t = X_{t-1}$.

Powers of the operators B and ∇ are defined as $B^j(X_t) = X_{t-j}$ and $\nabla^j(X_t) = \nabla(\nabla^{j-1}(X_t))$, with $\nabla^0 = X_t$. If the operator ∇ is applied to a linear trend function $m_t = c_0 + c_1 t$, then we obtain the constant function c_1, because $\nabla m_t = m_t - m_{t-1} = (c_0 + c_1 t) - (c_0 + c_1(t-1)) = c_1$. In the same way, any polynomial trend of degree k can be reduced to a constant by application of the operator ∇^k. These considerations suggest the possibility of, given any sequence $\{x_t\}$ of data, applying the operator ∇ repeatedly until we find a sequence that can plausibly be modelled as a realization of a stationary process. It is often found in practice that the order k of differencing required is quite small, frequently one or two.

Having a stationary time series, we will describe three models that are used to fit the data, namely: the moving average (MA), autoregressive (AR) and autoregressive moving average (ARMA) models.

Definition 2.2.1. $\{X_t\}$ is a moving average process of order q, $MA(q)$, if

$$X_t = \epsilon_t - \theta_1 \epsilon_{t-1} - ... - \theta_q \epsilon_{t-q}, q \in \mathbb{Z} \tag{2.11}$$

where $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and θ_1, ..., θ_q are constants.

An alternative way of writing 2.11 of model $MA(q)$ makes use of the backward shift operator B:

$$X_t = \theta(B)\epsilon_t, \tag{2.12}$$

with θ defined to be $\theta(B) = 1 - \theta_1 B - ... - \theta_q B^q$.

While the moving average process is a multiple regression with past errors as predictors (Hyndman and Athanasopoulos, 2014), in the autoregressive process, defined below, the predictors are lagged values $\{X_s : s < t\}$ of the time series $\{X_t\}$, plus a random component at instant t that is characterized to be white noise.

Definition 2.2.2. $\{X_t\}$ is an autoregressive process of order p, $AR(p)$, if

$$X_t = \phi_1 X_{t-1} + ... + \phi_p X_{t-p} + \epsilon_t \tag{2.13}$$

where $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and ϕ_1, ..., ϕ_p are constants, with $\phi_p \neq 0$ and $p \in \mathbb{N}$.

An alternative way of writing 2.13 of model $AR(p)$ is

$$\phi(B) X_t = \epsilon_t, \tag{2.14}$$

with ϕ defined to be $\phi(B) = 1 - \phi_1 B - ... - \phi_p B^p$.

We say that a time series is invertible if it has an autoregressive representation.

An autoregressive moving average process combines the two previous models, in the sense that it has as predictors both lagged values of the time series and also lagged errors.

Definition 2.2.3. $\{X_t\}$ is an autoregressive moving average process of orders p and q, $ARMA(p, q)$, if $\{X_t\}$ is stationary and if, for every t,

$$X_t = \phi_1 X_{t-1} + ... + \phi_p X_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - ... - \theta_q \epsilon_{t-q} \tag{2.15}$$

where $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and the polynomials $(1 - \phi_1 z - ... - \phi_p z^p)$ and $(1 - \theta_1 z - ... - \theta_p z^p)$ have no common factors.

An alternative way of writing 2.15 is given by:

$$\phi(B) X_t = \theta(B) \epsilon_t. \tag{2.16}$$

An important assumption of definition 2.2.3 is the requirement that $\{X_t\}$ is stationary. A stationary solution of the equation 2.15 exists (and is also the unique solution) if and only if $\phi(B)$ does not contain complex roots on the unitary circle, that is,

$$\phi(B) = 1 - \phi_1 B - ... - \phi_p B^p \neq 0, \text{ for all } |B| = 1. \tag{2.17}$$

The $ARMA(p, q)$ process can also be used to model time series with a seasonal component.

Definition 2.2.4. $\{X_t\}$ is an autoregressive process moving average of orders P and Q, with seasonal period S, i.e., $\{X_t\} \sim ARMA(P, Q)_S$, if, for every t,

$$\phi(B^S) X_t = \theta(B^S) \epsilon_t \tag{2.18}$$

where $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and the polynomials $\phi(z) = 1 - \phi_1 z - ... - \phi_p z^p$ and $\theta(z) = 1 - \theta_1 z - ... - \theta_p z^p$ have no common factors.

2.2.2 Models for Non-Stationary Time Series

The models previously presented are only applied for stationary time series. We shall introduce now the generalization of these models, that can be applied for non-stationary time series.

Definition 2.2.5. $\{X_t\}$ is an autoregressive integrated moving average of orders p, d and q, i.e., $\{X_t\} \sim ARIMA(p, d, q)$, if, for every t,

$$\phi(B)(1 - B)^d X_t = \alpha + \theta(B)\epsilon_t \tag{2.19}$$

where α is a real constant, $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and the polynomials $(1 - \phi_1 z - ... - \phi_p z^p)$ and $(1 - \theta_1 z - ... - \theta_p z^p)$ have no common factors.

Notice that ARIMA processes are reduced to ARMA processes when differenced finitely many times. If $d = 0$ then $\{X_t\}$ is a stationary time series.

Similarly to the $ARMA(p, q)$ process, the ARIMA process can also be used to model time series with a seasonal component.

Definition 2.2.6. $\{X_t\}$ is an autoregressive integrated moving-average of orders P and Q, with seasonal period S, i.e., $\{X_t\} \sim SARIMA(P, D, Q)_S$, if, for every t,

$$\Phi(B^S)(1 - B^S)^D X_t = \alpha + \Theta(B^S)\epsilon_t \tag{2.20}$$

where α is a real constant, $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and the polynomials $(1 - \Phi_1 z - ... - \Phi_p z^p)$ and $(1 - \Theta_1 z - ... - \Theta_p z^p)$ have no common factors.

The more general model combines the non-seasonal part of the model with the seasonal part of the model.

Definition 2.2.7. If d and D are non-negative integers, then $\{X_t\}$ is a $SARIMA(p, d, q) \times (P, D, Q)_S$ process if, for every t,

$$\Phi(B^S)\phi(B)(1 - B^S)^D(1 - B)^d X_t = \Theta(B^S)\theta(B)\epsilon_t \tag{2.21}$$

where $\{\epsilon_t\} \sim WN(0, \sigma_\epsilon^2)$ and the polynomials $(1-\phi_1 z-...-\phi_p z^p)$, $(1-\theta_1 z-...-\theta_p z^p)$, $(1-\Phi_1 z-...-\Phi_p z^p)$ and $(1 - \Theta_1 z - ... - \Theta_p z^p)$ have no common factors pairwise.

Having stated the existent models, given a time series, we need to identify the most adequate model and the respective orders that best fit the data. The necessary steps to properly define a model for the time series $\{X_t, t \in \mathbb{Z}\}$ given the observations $\{X_t, t = 1, ..., n\}$ are stated in the following enumeration (Hyndman and Athanasopoulos, 2014):

1. Plot the observations $\{X_t, t = 1, ..., n\}$ and choose suitable transformations in order to make the time series stationary. In particular, the first thing to do is to stabilize the variance. The logarithmic

transformation

$$T(X_t) = ln(X_t) \tag{2.22}$$

can be appropriate whenever $\{X_t\}$ is a series whose standard deviation increases linearly with the mean. If the linearity is not clear, then a general class of variance-stabilizing transformations, namely the the Box-Cox transformation, should be applied, with defining equation

$$T_\lambda(X_t) = \begin{cases} \frac{X_t^\lambda - 1}{\lambda} & , \lambda \neq 0 \\ ln(X_t) & , \lambda = 0 \end{cases}$$

where, usually, λ takes values in the interval $[-1, 1]$.

After the variance is stabilized, it is necessary to check whether the time series is stationary. If not, then the mean should be stabilized too, that is, the trend should be eliminated, as well as the seasonality, if any, by taking differences.

- The seasonal backward shift operator with period S of order D, $(1 - B^S)^D$, should be used to eliminate seasonalities of period S, where usually $D \leq 1$;
- The simple backward shift operator of order d, $(1 - B)^d$, should be used to eliminate trends without a seasonal component, where usually $d \leq 2$;

2. Once the data is stationary, we are faced with the problem of selecting appropriate values for the orders p, q, P and Q. The right identification comes from examining the ACF and PACF values. The orders P and Q should be selected by analysing the values of ACF and PACF at lags that are multiple of S (S, $2S$, $3S$,...), whereas p and q should be selected by analysing the values of ACF and PACF at lags 1, 2,..., S-1, according to Table 2.1 (Pacheco, 2001).

	ACF $\{\rho_h\}$	**PACF** $\{\phi_{hh}\}$
$AR(p)$	Exponential or sinusoidal decay to zero	=0, $k > p$
$MA(q)$	=0, $k > q$	Exponential or sinusoidal decay to zero
$ARMA(p, q)$	Exponential or sinusoidal decay to zero from the order q+1	Exponential or sinusoidal decay to zero from the order p+1

Table 2.1: Auxiliary Table for the choice of the orders p and q.

3. If the previous step results in competing alternatives of SARIMA models with different orders, the information criterion of Akaike, known as the AIC, can be used in order to select the best one. Here we define a bias-corrected version of the AIC, referred to as the AICC, given by

$$AICC(\beta) = -2ln(L_X(\beta, S_X^2, S_\epsilon^2)) + 2(p + q + 1)n/(n - p - q - 2). \tag{2.23}$$

In equation 2.23, L is the Normal likelihood for an ARMA process, β is the coefficient vector of the fitted model and S_X^2 and S_ϵ^2 are the sample variance of the time series and of the residuals, respectively. We therefore select the values of p and q for the fitted model to be those that minimize

$AICC(\hat{\beta})$, where $\hat{\beta}$ is the estimated coefficient vector of the fitted model that can be computed by methods such as the maximum likelihood or the minimum squares.

4. Having chosen the model, we check for the goodness of fit, which is judged by comparing the observed values with the corresponding predicted values obtained from the fitted model. Let the residuals be the difference between the observed values and the fitted values. If the fitted model is appropriate, then the residuals should be white noise and, in particular, have the following properties:

   - The random variables $\{\epsilon_t\}$ should be normally distributed, property that can be checked by analysing the histogram or the Q-Q plot of the time series;
   - The mean of the residuals should be zero and the variance should be constant, that is, the time series of the residuals should be stationary around zero, which can be checked either by analysing the plot of the residuals or by performing a Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test, whose null hypothesis is that the data are stationary and non-seasonal;
   - $\{\epsilon_t\}$ should be uncorrelated, which can be checked by performing a *Portmanteau* test, which is a class of tests characterized by involving the ACF the residuals. If $\{\rho_k(\epsilon) : k = 0, 1, ...\}$ are the ACF values of the residuals of the model and $m = max(p, q)$, then we test

   $$H_0 : \rho_1(\epsilon) = ... = \rho_m(\epsilon) = 0 \text{ vs. } H_1 :\sim H_0. \tag{2.24}$$

   One of the tests belonging to this class was developed by Ljung and Box and has the following statistic:

   $$Q = n(n-2)\sum_{k=1}^{m} \frac{1}{n-k}\hat{\rho_k}^2(\hat{\epsilon}) \tag{2.25}$$

   whose distribution is better approximated by the chi-squared distribution with $m-p-q$ degrees of freedom, under H_0. A large value of Q suggests that the sample autocorrelations of the data are too large for the data to be a sample from an i.i.d. sequence. We therefore reject the i.i.d. hypothesis at level α if $Q > \chi^2_{1-\alpha}(m-p-q)$, where $\chi^2_{1-\alpha}(m-p-q)$ is the $1-\alpha$ quantile of the chi-squared distribution with $m-p-q$ degrees of freedom.

   If the stated properties are not satisfied, then there is information left in the residuals that should be used in computing forecasts, which means that we need to go back to step 2 and find a more appropriate model that fits the data.

5. Once the residuals look like white noise, calculate forecasts of future values. This step will be detailed in the next Section (Section 2.3).

2.3 Forecast

In this Section, we develop, according to Wei (2006), the minimum square error forecasts for a time series that follows a stationary (Subsection 2.3.1) or non-stationary (Subsection 2.3.2) time series model.

2.3.1 Forecast Stationary Time Series

To derive the minimum mean square error forecast, we start by considering the stationary ARMA model

$$\phi(B)X_t = \theta(B)\epsilon_t. \tag{2.26}$$

Because the model is stationary, we can rewrite it in a moving average representation

$$X_t = \psi(B)\epsilon_t = \epsilon_t + \psi_1\epsilon_{t-1} + \psi_2\epsilon_{t-2} + ... \tag{2.27}$$

where

$$\psi(B) = \frac{\theta(B)}{\phi(B)} = \sum_{j=0}^{\infty}\psi_j B^j \tag{2.28}$$

and $\psi_0 = 1$. For $t = n + h$, we have

$$X_{n+h} = \sum_{j=0}^{\infty}\psi_j\epsilon_{n+h-j}. \tag{2.29}$$

Suppose that at time $t = n$, we have the observations $X_n, X_{n-1}, X_{n-2}, ...$ and wish to forecast h-step ahead of future value X_{n+h} as a linear combination of the observations $X_n, X_{n-1}, X_{n-2}, ...$ Because X_t for $t = n, n-1, n-2, ...$ can all be written in the form 2.27, we can let the minimum square error forecast $\hat{X}_n(h)$ of X_{n+h} be

$$\hat{X}_n(h) = \psi_h^*\epsilon_n + \psi_{h+1}^*\epsilon_{n-1} + \psi_{h+2}^*\epsilon_{n-2} + ... \tag{2.30}$$

where the ψ_j^* are to be determined. The mean square error of the forecast is

$$\begin{aligned} E[(X_{n+h} - \hat{X}_n(h))^2] &= Var(X_{n+h} - \hat{X}_n(h)) + E[X_{n+h} - \hat{X}_n(h)]^2 \\ &= Var\left(\sum_{j=0}^{\infty}\psi_j\epsilon_{n+h-j} - (\psi_h^*\epsilon_n + \psi_{h+1}^*\epsilon_{n-1} + \psi_{h+2}^*\epsilon_{n-2} + ...)\right) + 0 \\ &= \sigma_\epsilon^2\sum_{j=0}^{h-1}(\psi_j^2) + \sigma_\epsilon^2\sum_{j=0}^{\infty}(\psi_{h+j} - \psi_{h+j}^*), \end{aligned} \tag{2.31}$$

which is minimized when $\psi_{h+j}^* = \psi_{h+j}$. Hence,

$$\hat{X}_n(h) = \psi_l\epsilon_n + \psi_{h+1}\epsilon_{n-1} + \psi_{h+2}\epsilon_{n-2} + ... \tag{2.32}$$

Note that

$$E[\epsilon_{n+h}|X_n, X_{n-1}, ...] = \begin{cases} 0 & , h > 0 \\ \epsilon_{n+h} & , h \leq 0 \end{cases}$$

because, for $h > 0$, ϵ_{n+h} is independent of $X_n, X_{n-1}, ...$, due to the fact that $\{\epsilon_t\}$ is white noise, and, for $h \leq 0$, ϵ_{n+h} is deterministic, due to the moving average representation of $\{X_t\}$.

The minimum square error forecast is given by its conditional expectation. Using 2.29 and the con-

ditional expected value of the error terms, we reach the formula for $\hat{X}_n(h)$, the h-step ahead forecast of X_{n+h} at the forecast origin n:

$$\hat{X}_n(h) = E[X_{n+h}|X_n, X_{n-1}, ...] = \psi_h\epsilon_n + \psi_{h+1}\epsilon_{n-1} + \psi_{h+2}\epsilon_{n-2} + ... \tag{2.33}$$

The forecast error is

$$e_n(h) = X_{n+h} - \hat{X}_n(h) = \sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j}. \tag{2.34}$$

Since $E[e_n(h)|X_t, t \leq n] = 0$, the forecast is unbiased with error variance

$$Var(e_n(h)) = \sigma_\epsilon^2 \sum_{j=0}^{h-1} \psi_j^2. \tag{2.35}$$

If $\{X_t, t \in \mathbb{N}\}$ is a normal process then we conclude that, for $0 < \alpha < 1$,

$$P\left(-N_{\alpha/2} < \frac{X_{t+h} - \hat{X}_n(h)}{\sigma_\epsilon \sqrt{\sum_{j=0}^{h-1} \psi_{j=0}^2}} < N_{\alpha/2}\right) = 1 - \alpha \tag{2.36}$$

where N is the standard normal random variable and $N_{\alpha/2}$ is such that $P(N > N_{\alpha/2}) = \alpha/2$, so the forecast limits are

$$\hat{X}_n(h) \pm N_{\alpha/2}\sigma_\epsilon \sqrt{\sum_{j=0}^{h-1} \psi_j^2}. \tag{2.37}$$

2.3.2 Forecast Non-Stationary Time Series

Let us consider the general non-stationary $ARIMA(p, d, q)$ model with $d \neq 0$, i.e.,

$$\Phi(B)(1 - B)^d X_t = \Theta(B)\epsilon_t. \tag{2.38}$$

We rewrite the model at time $t+l$ in an AR representation (that exists because the model is invertible). Thus,

$$\pi(B)X_{t+h} = \epsilon_{t+h}, \tag{2.39}$$

where

$$\pi(B) = \frac{\Phi(B)(1 - B)^d}{\Theta(B)} = 1 - \sum_{j=1}^{\infty} \pi_j B^j. \tag{2.40}$$

The h-step ahead forecast of X_{n+l} at the forecast origin n is given by (see Wei [2006] for the full proof):

$$\hat{X}_n(h) = E[X_{n+h}|X_n, X_{n-1}, ...] = \sum_{j=1}^{\infty} \pi_j^{(h)} X_{n-j+1}, \tag{2.41}$$

where $\pi_j^{(h)} = \sum_{i=0}^{h-1} \pi_{h-1+j-i}\phi_i$, and the forecast error is

$$e_n(h) = X_{n+h} - \hat{X}_n(h) = \sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j}, \tag{2.42}$$

where the weights ψ_j can be calculated recursively from the π_j weights as $\psi_j = \sum_{i=0}^{j-1} \pi_{j-1}\psi_i$, for $j = 1, ..., h-1$.

The forecast error is unbiased and we can obtain forecast confidence intervals using the same reasoning as for the stationary processes.

2.3.3 Evaluate Forecast Accuracy

In order to evaluate the forecast accuracy, let $\{e_k(1) = X_{k+1} - \hat{X}_k(1) : k = j+1, j+2, ..., n-1\}$ be the one-step ahead forecast errors. The method that is mostly used throughout the thesis when comparing forecasts using different models will be the Mean Absolute Percentage Error ($MAPE$), which is computed as

$$MAPE = \frac{1}{n-j} \sum_{k=j}^{n-1} \left| \frac{e_k(1)}{X_{k+1}} \right|, \tag{2.43}$$

For a given model, the closer to zero is its $MAPE$ value, the better is the adequacy of the model. The advantage of this measure is that it is scale independent. On the other hand, one disadvantage is that it is only sensible if $X_t \neq 0$ for all t.

2.4 Generalized Additive Models

In this Section, we introduce a type of models, Generalized Additive Models (GAMs), that incorporate explanatory variables. Suppose that we have a response random variable Y and a set of p predictor random variables $W_1, W_2, ..., W_p$. A set of n independent realizations of these random variables is denoted by $(y_1, w_{11}, ..., w_{1p}), ..., (y_n, w_{n1}, ..., w_{np})$.

Before introducing GAMs, we start by defining Generalized Linear Models (GLMs), which, according to Wood [2006], are linear models that allow for response distributions other than Normal and a degree of non-linearity in the model structure. A GLM has the basic structure

$$g(\mu_i) = \alpha_0 + \alpha_1 W_1 + ... + \alpha_p W_p, \tag{2.44}$$

where $\mu_i = E[Y_i]$, g is a smooth (i.e., continuous and differentiable) monotonic link function, W_i is the ith row of a model matrix W and $\alpha_0, \alpha_1, ..., \alpha_p$ are unknown parameters. In addition, a GLM makes the distribution assumptions that the Y_i are independent and distributed with some exponential family distribution. The exponential family of distributions includes many distributions that are useful for practical modelling, such as the Poison, Binomial, Gamma and Normal distributions.

A Generalized Additive Model is a Generalized Linear Model with a linear predictor involving a sum

of smooth functions of covariates,

$$g(\mu_i) = f_0 + f_1(W_1) + ... + f_p(W_p), \tag{2.45}$$

where f_j are smooth functions of the covariates W_j for all $j = 0, ..., p$ and $W_0 = I$. Subsections 2.4.1, 2.4.2 and 2.4.3 are intended to explain the methods on how to select these smooth functions.

2.4.1 Representing smooth functions by regression splines

For simplicity, let us look at the case of a single predictor and where the link function is the identity function:

$$Y = f(W). \tag{2.46}$$

Consider n observations (w_j, y_j), $j = 1, ..., n$, of covariates and dependent variables. In order to estimate f, this function is represented in such a way that equation 2.46 becomes a linear model. This can be done by choosing a basis, that is, by defining the space of functions of which f is an element.

So, we introduce the concept of spline curve, which is a piecewise polynomial curve, i.e., it joins two or more polynomial curves and the locations of the joins are known as knots. If $b_i(x)$ is the ith such basis function, then the smooth function f can be represented as

$$f(w) = \sum_{i=1}^{q} \beta_i b_i(w) = \beta^T b(w) \tag{2.47}$$

where β_i are unknown parameters (the spline coefficients), β is a vector containing the spline coefficients and $b(w)$ is the vector containing the spline basis functions. We say that f is modelled by regression splines. Substituting equation 2.47 into 2.46 yields a linear model. Examples of basis are the polynomial basis or the cubic spline basis.

2.4.2 Controlling the degree of smoothing with penalized regression splines

The model's smoothness can be controlled by adding a penalty to the least squares fitting objective, known as smoothing parameter, λ. For example, rather than minimizing the residual sum of squares

$$||y - \beta^T B(w)||^2, \tag{2.48}$$

where B is the matrix with the rows $b(w_1)^T, b(w_2)^T, ..., b(w_n)^T$ containing the evaluated spline basis functions, we instead add to the residual sum of squares a penalty: some multiple λ of the integral of the squared second derivative of $f(w)$ with respect to w, which penalizes steep slopes. Consider a small interval δw over which the second derivative $f''(w)$ of the smoother $f(w)$ is approximately constant. The contribution of that interval to the penalty is then $\lambda f''(w)^2 \delta w$.

Then it is minimized

$$||y - \beta^T B(w)||^2 + \lambda \int_0^1 [f''(w)]^2 dw, \tag{2.49}$$

where the constant λ is determined by cross correlation (detailed in Section 2.4.3). If instead of a single variable, we have several variables, then the contributions of several variables can be added. There is then one λ_i, $i = 1, ..., p$, for each of the p variables.

The trade-off between model fit and model smoothness is controlled by λ in the sense that $\lambda \to \infty$ leads to a straight line estimate for f, while $\lambda = 0$ results in an unpenalized regression spline estimate. Because f is linear in the parameters β_i, the penalty can always be written as a quadratic form in β,

$$\int_0^1 [f''(w)]^2 dw = \beta^T S \beta, \tag{2.50}$$

where S is a matrix of known coefficients. Therefore, the penalized regression spline fitting problem is to minimize

$$||y - \beta^T B(w)||^2 + \lambda \beta^T S \beta, \tag{2.51}$$

with respect to β.

It can be shown (see Wood [2006]) that the formal expression for the minimizer of equation 2.51, the penalized least squares estimator of β, is

$$\hat{\beta} = (B^T B + \lambda S)^{-1} B^T y \tag{2.52}$$

and the hat matrix A of the model can be written as

$$A = B(B^T B + \lambda S)^{-1} B^T. \tag{2.53}$$

The problem of estimating the degree of smoothness for the model is now the problem of estimating the smoothing parameter λ, which can be done by cross validation.

2.4.3 Choosing the smoothing parameter λ using cross validation

We want to choose λ so that the spline estimate $\hat{f}$ is as close as possible to f. A suitable criterion might be to choose λ to minimize

$$M = \frac{1}{n} \sum_{i=1}^{n} (\hat{f}_i - f_i)^2, \tag{2.54}$$

where $\hat{f}_i \equiv \hat{f}(x_i)$ and $f_i \equiv f(x_i)$. Since f is unknown, M cannot be used directly, but it is possible to derive an estimate of $E[M] + \sigma^2$, which is the expected squared error in predicting a new variable. Let $\hat{f}^{[-i]}$ be the model fitted to all data except y_i, and define the Ordinary Cross Validation (OCV) score

$$\mathcal{V}_0 = \frac{1}{n} \sum_{i=1}^{n} (\hat{f}^{[-i]} - y_i)^2. \tag{2.55}$$

This score results from leaving out each datum in turn, fitting the model to the remaining data and calculating the squared difference between the missing datum and its predicted value: these squared

differences are then average over all data. Substituting $y_i = f_i + \epsilon_i$,

$$\mathcal{V}_0 = \frac{1}{n}\sum_{i=1}^{n}(\hat{f}^{[-i]} - f_i - \epsilon_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(\hat{f}^{[-i]} - f_i)^2 - 2(\hat{f}^{[-i]} - f_i)\epsilon_i + \epsilon_i^2. \tag{2.56}$$

Since $E[\epsilon_i] = 0$ and ϵ_i and $\hat{f}^{[-i]}$ are independent, the second term in the summation vanishes if expectations are taken, leading to

$$E[\mathcal{V}_0] = \frac{1}{n}E\left[\sum_{i=1}^{n}(\hat{f}^{[-i]} - f_i)^2\right] + \sigma^2. \tag{2.57}$$

Now $\hat{f}^{[-i]} \simeq \hat{f}$ with equality in the large sample limit, so $E[\mathcal{V}_0] \simeq E[M] + \sigma^2$ also with equality in the large sample limit. Hence, we choose λ in order to minimize $\mathcal{V}_0$, which is known as ordinary cross validation.

It is inefficient to calculate $\mathcal{V}_0$ by leaving out one datum at a time, and fitting the model to each of the n resulting data sets, but it can be shown that

$$\mathcal{V}_0 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_i)^2/(1 - A_{ii})^2, \tag{2.58}$$

where $\hat{f}$ is the estimate from fitting to all the data and A is the corresponding hat matrix. In practice, the weights $1 - A_{ii}$ are often replaced by the mean weight $tr(I - A)/n$, in order to arrive at the Generalized Cross Validation (GCV) score

$$\mathcal{V}_g = \frac{n\sum_{i=1}^{n}(y_i - \hat{f}_i)^2}{(tr(I - A))^2}, \tag{2.59}$$

which has computational advantages over OCV. Therefore, we choose λ to minimize $\mathcal{V}_g$.

Having selected the best λ, it is possible to compute $\hat{\beta}$ and, hence, we reach the model that best fits the data. Having new values for the covariates, we can use the model to compute new values of the response variable.

2.5 Clustering Time Series

Clustering refers to a very broad set of techniques for finding subgroups in a data set. When we cluster the observations of a data set, we seek to partition them into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other (James et al. 2015). A crucial question in cluster analysis is establishing what we mean by similar data objects, i.e., determining a suitable similarity/dissimilarity measure between two objects. In the specific context of time series data, the concept of dissimilarity is particularly complex due to the dynamic character of the series. Dissimilarities usually considered in conventional clustering could not work adequately with time dependent data because they ignore the interdependence relationship between values (Montero and Vilar 2014). Considering this, it can be applied hierarchical clustering, PAM (Partitioning Around Medoids) algorithm or DBSCAN for clustering time series. Throughout the

thesis, our results are based on the hierarchical algorithm, so this is the approach that we will fully describe.

2.5.1 Hierarchical Clustering

Hierarchical clustering is one of the best-known clustering approaches. In hierarchical clustering, we do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a dendrogram, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to m, where m is the total number of time series that we consider for clustering.

We will describe bottom-up or agglomerative clustering, according to James et al., 2015. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk. We will begin with a discussion of how to interpret a dendrogram and then discuss how hierarchical clustering is actually performed.

Each leaf of the dendrogram represents one of the m observations. As we move up the tree, some leaves begin to fuse into branches. These correspond to observations that are similar to each other. As we move higher up the tree, branches themselves fuse, either with leaves or other branches. The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other. On the other hand, observations that fuse later (near the top of the tree) can be quite different. The height of the cut in the dendrogram controls the number of clusters obtained. The term hierarchical refers to the fact that clusters obtained by cutting the dendrogram at a given height are necessarily nested within the clusters obtained by cutting the dendrogram at any greater height.

We now describe the algorithm of hierachical clustering. We begin by defining some sort of dissimilarity measure between each pair of observations. The algorithm proceeds iteratively. Starting out at the bottom of the dendrogram, each of the m observations is treated as its own cluster. The two clusters that are most similar to each other are then fused so that there are now $m-1$ clusters. Next, the two clusters that are most similar to each cluster are fused again, so that there are now $m-2$ clusters. The algorithm proceeds in this fashion until all of the observations belong to one single cluster, and the dendrogram is complete (algorithm 1).

Algorithm 1 Hierarchical Clustering Algorithm

1. Begin with m observations and a measure of all the $\binom{m}{2}$ pairwise dissimilarities. Treat each observation as its own cluster.
2. For $i = m, m-1, ..., 2$:
   - Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
   - Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

The notion of linkage defines the dissimilarity between two groups of observations. The three most

common types of linkage are complete, average and single, defined as follows: given two clusters A and B, complete (resp. average, single) linkage records the largest (resp. average, smallest) value of all pairwise dissimilarities between the observations in cluster A and the observations in cluster B. Average and complete linkage are generally preferred over single linkage, as they tend to yield more balanced dendrograms.

2.5.2 Choice of dissimilarity measure

The choice of dissimilarity measure is very important, as it has a strong effect on the resulting dendrogram. We took into account several dissimilarity measures (see Montero and Vilar 2014), but we end up choosing the one based on the spectral analysis of a time series proposed by Caiado et al. [2014].

According to Brockwell and Davis [2002], the spectral representation of a stationary time series $\{X_t\}$ essentially decomposes $\{X_t\}$ into a sum of sinusoidal components with uncorrelated random coefficients. In conjunction with this decomposition there is a corresponding decomposition into sinusoids of the autocovariance function of $\{X_t\}$. The spectral decomposition is thus an analogue for stationary processes of the more familiar Fourier representation of deterministic functions. The analysis of stationary processes by means of their spectral representation is often referred to as the "frequency domain analysis" of time series or "spectral analysis".

Suppose that $\{X_t\}$ is a zero-mean stationary time series with autocovariance function γ satisfying $\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty$. The spectral density of $\{X_t\}$ is the function f defined by

$$f(\lambda) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} e^{-ih\lambda}\gamma(h), \tag{2.60}$$

where $e^{i\lambda} = cos(\lambda) + isen(\lambda)$ and $i = \sqrt{-1}$.

To introduce the periodogram, we consider the vector of complex numbers $\boldsymbol{x} = \{x_1, ..., x_n\}^T$. Now let

$$\omega_k = \frac{2\pi k}{n}, k = -\frac{n-1}{2}, ..., \frac{n}{2}. \tag{2.61}$$

We shall refer to the set F_n of the values $\{\omega_k\}$ as the Fourier frequencies associated with sample size n, noting that F_n is a subset of the interval $(-\pi, \pi]$. Correspondingly, we introduce the n vectors

$$\boldsymbol{e_k} = \frac{1}{\sqrt{n}} \begin{bmatrix} e^{i\omega_k} \\ e^{2i\omega_k} \\ ... \\ e^{ni\omega_k} \end{bmatrix}, k = -\frac{n-1}{2}, ..., \frac{n}{2}. \tag{2.62}$$

Note that $\boldsymbol{e_1}, ..., \boldsymbol{e_n}$ are orthonormal. This implies that $\boldsymbol{e_1}, ..., \boldsymbol{e_n}$ is a basis for $\mathbb{C}_n$, so that any $\boldsymbol{x} \in \mathbb{C}_n$ can be expressed as the sum of n components,

$$\boldsymbol{x} = \sum_{k=-(n-1)/2}^{n/2} \boldsymbol{a}_k \boldsymbol{e}_k. \tag{2.63}$$

and, in particular, the tth component of $\boldsymbol{x}$ is given by

$$x_t = \sum_{k=-(n-1)/2}^{n/2} a_k(cos(\omega_k t) + isen(\omega_k t)), t = 1, ..., n, \tag{2.64}$$

showing that 2.63 is just a way of representing x_t as a linear combination of sine waves with frequencies $\omega_k \in F_n$.

By manipulating 2.63, we get the expression for the coefficients a_k:

$$a_k = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} x_t e^{-it\omega_k}. \tag{2.65}$$

The sequence $\{a_k\}$ is called the discrete Fourier transform of the sequence $\{x_1, ..., x_n\}$.

We are now able to define the periodogram of a time series

Definition 2.5.1. The periodogram of $\{x_1, ..., x_n\}$ is the function

$$I_n(\lambda) = \frac{1}{n} \left| \sum_{t=1}^{n} x_t e^{-it\lambda} \right|^2. \tag{2.66}$$

The frequency domain approach allows to measure the dissimilarity between time series. The key idea is to assess the dissimilarity between the corresponding spectral representations of the series, which is the base of periodogram-based distances as explained below.

Let $I_{X_T}(\lambda_k) = T^{-1}|\sum_{t=1}^{T} X_t e^{-i\lambda_k t}|^2$ and $I_{Y_T}(\lambda_k) = T^{-1}|\sum_{t=1}^{T} Y_t e^{-i\lambda_k t}|^2$ be the periodograms of X_T and Y_T, respectively, at frequencies $\lambda_k = 2\pi k/T$, $k = 1, ..., n$, with $n = (T-1)/2$. If we are not interested in the process scale but only on its correlation structure, better results can be obtained using the Euclidean distance between the normalized periodogram ordinates. Besides that, as the variance of the periodogram ordinates is proportional to the spectrum value at the corresponding frequencies, it makes sense to use the logarithm of the normalized periodogram:

$$d_{NLP}(X_T, Y_T) = \frac{1}{n} \sqrt{\sum_{k=1}^{n} (logNI_{X_T}(\lambda_k) - logNI_{Y_T}(\lambda_k))^2},$$

where $NI_{X_T} = I_{X_T}(\lambda_k)/\hat{\gamma_{0,X_T}}$ and $NI_{Y_T} = I_{Y_T}(\lambda_k)/\hat{\gamma_{0,Y_T}}$, with $\hat{\gamma_{0,X_T}}$ and $\hat{\gamma_{0,Y_T}}$ being the sample variances of X_T and Y_T, respectively.

2.5.3 Validating the Number of Clusters

Having obtained a partition from the clustering, it remains to verify whether the clusters that have been found represent true subgroups in the data.

A wide variety of indexes have been proposed to find the optimal number of clusters in a partitioning of a data set during the clustering process (see Charrad et al., 2014). We used mainly 4 indexes (Gamma, Silhouette, Dunn and Entropy) that we proceed to explain. Let n be the number of observations.

The Gamma index represents an adaptation of Goodman and Kriskal's Gamma statistic for use in clustering situation. Comparisons are made between all within-cluster dissimilarities and all between-cluster dissimilarities. A comparison is considered to be concordant $[s(+)]$ (resp. discordant $[s(-)]$) if a within-cluster dissimilarity is strictly less (resp. strictly greater) than a between-cluster dissimilarity, equalities between members of two sets of dissimilarities are disregarded in the definition of the index.

$$Gamma = \frac{s(+) - s(-)}{s(+) + s(-)},$$

where $s(+)$ is the number of concordant comparisons and $s(-)$ is the number of discordant comparisons. The maximum value of the index is taken to represent the correct number of clusters.

The silhouette index is given by

$$Silhouette = \frac{\sum_{i=1}^{n} S(i)}{n},$$

where

- $S(i) = \frac{b(i)-a(i)}{max\{a(i),b(i)\}}$;
- $a(i) = \frac{\sum_{j \in \{C_r \setminus i\}} d_{ij}}{n_r - 1}$ is the average dissimilarity of the ith object to all other objects of cluster C_r;
- $b(i) = min_{s \neq r}\{d_{iC_s}\}$;
- $d_{iC_s} = \frac{\sum_{j \in C_s} d_{ij}}{n_s}$ is the average dissimilarity of the ith object to all objects of cluster C_s.

The maximum value of the index is used to determine the optimal number of clusters in the data.

The Dunn index defines the ratio between the minimal intercluster distance to maximal intracluster distance. This index is given by

$$Dunn = \frac{min_{1 \leq i < j \leq q} d(C_i, C_j)}{max_{1 \leq k \leq q} diam(C_k)}, \tag{2.67}$$

where $d(C_i, C_j)$ is the dissimilarity function between two clusters C_i and C_j defined as $d(C_i, C_j) = min_{x \in C_i, y \in C_j} d(x, y)$ and $diam(C)$ is the diameter of a cluster, which may be considered as a measure of cluster dispersion and can be defined by $diam(C) = max_{x,y \in C} d(x, y)$. If the data set contains compact and well-separated clusters, the diameter of the cluster is expected to be small and the distance between clusters is expected to be large. Thus, Dunn index should be maximized.

Finally, the index entropy is the degree to which each cluster consists of objects of a single class. For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the probability that a member of cluster i belongs to class j as $p_{ij} = m_{ij}/m_i$, where m_i is the number of objects in cluster i and m_{ij} is the number of objects of class j in cluster i. Using this class distribution, the entropy of each cluster i is calculated using the formula

$$e_i = -\sum_{j=1}^{L} p_{ij} log_2 p_{ij},$$

where L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the

entropies of each cluster weighted by the size of each cluster, i.e.,

$$e = \sum_{i=1}^{K} \frac{m_i}{m} e_i,$$

where K is the number of clusters and m is the total number of data points. The optimal number of clusters is chosen to be the number that returns an entropy value that is closer to zero.

Chapter 3

Results and Discussion

This Chapter is dedicated to state the results of clustering 25 load curves and forecasting one day ahead electricity consumption of the representative time series of each cluster. Firstly, Section 3.1 is intended to describe the given data for this study. In Section 3.2 it is presented the clustering of the time series given. Finally, in Sections 3.3 and 3.4 it is explained the procedure of fitting a Generalized Additive Model and a SARIMA model, respectively, to the representative time series of the cluster, as well as presenting the results of the forecast.

3.1 Description of the Data

We were given 25 time series, each containing observations of electrical consumption of a certain customer at every XXX minutes, measured in kilowatt (kW), starting at 2015-05-05 05:15 and ending at 2017-03-19 09:00, which accounts for almost two years of observations. Figure 3.1 contains the plot of these 25 time series, labelled from 1 to 25. We can immediately see from this Figure that time series 18 stands out from the other ones, because it has a completely different scale, with measurements between, approximately, XXX kW and XXX kW. The remaining time series have measurements that do not go beyond XXX kW.

The first step before performing a further analysis of the time series is to run an algorithm that, for each time series, identifies observations that are outliers and replaces them by linear interpolation. Note that the main objective of this study is not the detection of outliers, so we used one of the simplest method to detect outliers, which is the Tukey's method.

Besides 25 load curves, we were also given weather time series, because, for the modelling using GAM, we intend to include weather variables in the model. Each of the 25 time series is from a different location, and, for each location, we were given 7 weather variables, namely, the radiance, atmospheric pressure, temperature, wind direction, wind speed, wind U and wind V (where a positive U component represents wind blowing to the East and a positive V represents wind to the North).

Figure 3.1: Confidential annex: Plot of each 25 time series used for clustering.

The time period for which we have information about the weather is between 2015-05-05 at 05:15 and 2017-03-19 at 09:00, the same period as for the electrical consumption time series. For each day within this period, at 00:00, it is recorded predictions of the measurements of the weather variables at every 3 hours for 3 days ahead. We only considered 1 day ahead predictions throughout this study, so, for every day, we have weather predictions at 00:00, 03:00, and so on, until 21:00.

To have an idea of the weather variables we are dealing with, we choose a random load time series in order to plot its associated weather variables. Figure 3.2 contains plots of each weather variable associated with the load time series 16. From this Figure, we can see, for instance, that variables temperature and radiance are highly related with each other, since they present a similar pattern (this is concluded by eye now, whereas in Section 3.3 we will detail and prove this assertion with auxiliary graphics and values for the correlation). Beside this relation, it is clear that all weather time series have in common a period (around the two first weeks of August 2016) with missing values, which were replaced using interpolation.

Notice that weather observations are measured at every 3 hours, whereas the electricity consumption is measured at every XXX minutes. In order to fit a GAM, the measurements of the variables must have the same granularity. So, our aim is to obtain measures for the weather at every XXX minutes as well. We considered the following two methods in order to disaggregate low frequency time series into higher frequency series:

1. interpolation;

2. fitting a regression model to each weather time series and getting the desired values of granularity from the model (Denton 1971).

In this latter approach, temporal disaggregation can be performed with or without one or more high frequency indicator series, which can be seen as predictor variables. However, when there is no indicator series, which is the case, the accuracy of the resulting high frequency series will be low. So, the disaggregation was performed using simple interpolation.

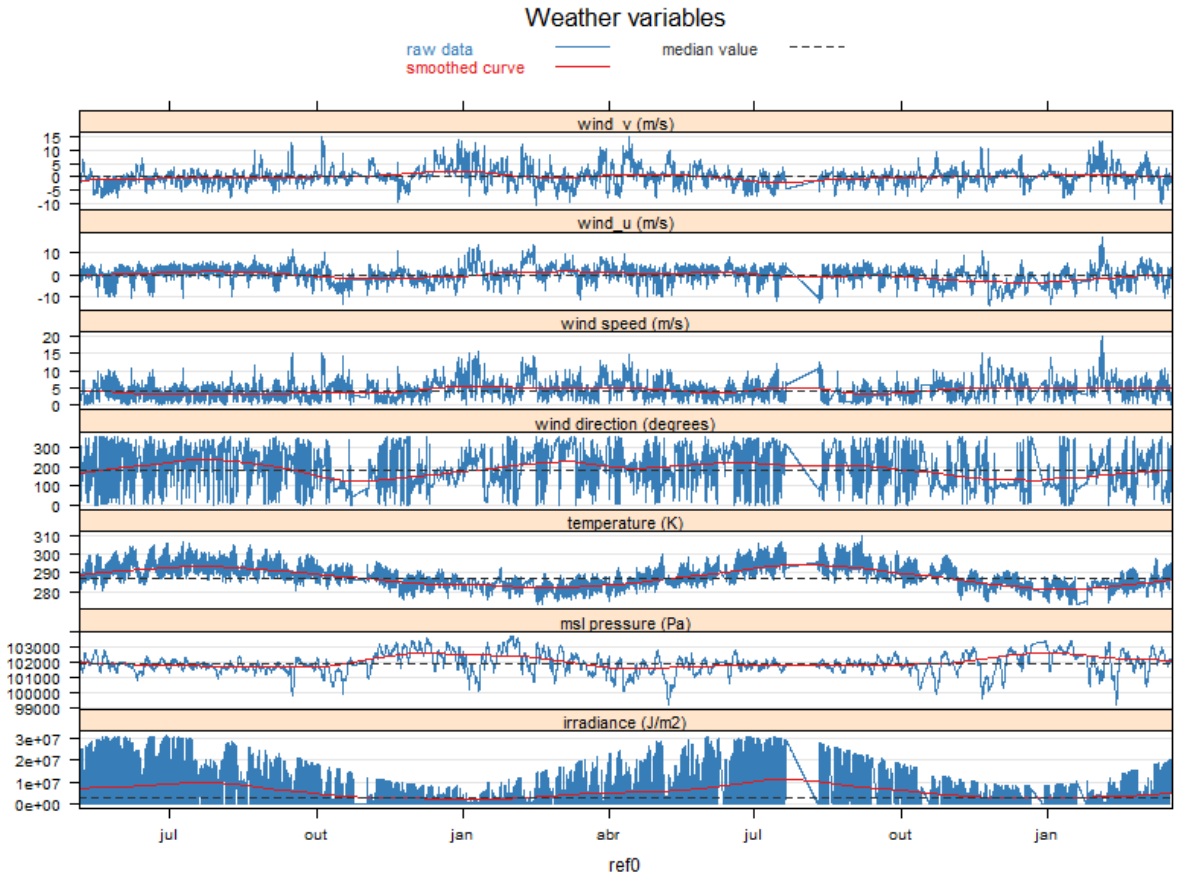

Figure 3.2: Weather variables associated with the load time series 16.

In the next Section it is detailed the clustering procedure, which only makes use of the 25 load time series, clean of outliers. Weather time series, clean of outliers as well, will be used in Section 3.3, when fitting a generalized additive model.

3.2 Time Series Clustering

This Section is intended to state the results of grouping 25 load time series by similarity of consumption. In order to output this grouping, it is necessary to indicate what is the definition of similarity of consumption. First of all, note that, when plotting each of the 25 load time series, we could understand that these have different scales. This means that, if we perform the clustering without any transformation of the time series, these will be grouped by similarity of consumption in scale. We will call this approach of clustering as approach 1 (represented in Figure 3.3). In this case, for instance, time series 18, which is the one that stands out from Figure 3.1, would be placed alone in a cluster, as expected. If the clustering is performed again without time series 18, then we get the dendrogram of Figure 3.4, partitioned in 4 clusters, which was the best number of clusters returned by the validation indexes.

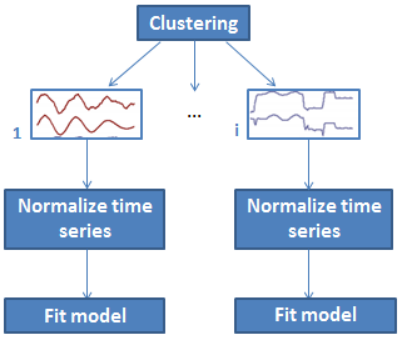

Figure 3.3: Approach 1 for clustering and fitting models for i clusters.

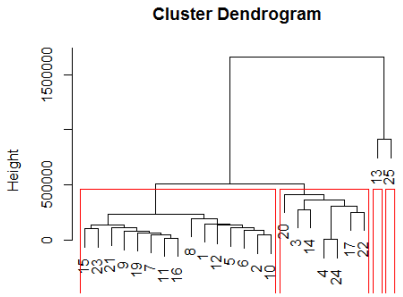

Figure 3.4: Dendrogram for approach 1, using hierarchical clustering with CORT distance and average agglomeration method.

The final clusters of approach 1 are $C_1 = \{18\}$, $C_2 = \{25\}$, $C_3 = \{13\}$, $C_4 = \{3, 4, 14, 17, 20, 22, 24\}$, and $C_5 = \{1, 2, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 19, 21, 23\}$. Paying attention to the scales, we notice that the maximum values that the time series within clusters 1, 2, 3, 4 and 5 take are around XXX, XXX, XXX, XXX and XXX kW, respectively. That is, the scale is the factor that is being considering when grouping time series, as mentioned earlier.

However, when clustering time series, we decided that we should give primarily attention to group customers with the same consumption pattern rather than focusing on scale, because we want to fit models that clearly translate daily, weekly or yearly seasonalities. So, first, it is necessary to normalize each time series. We will call this approach of clustering as approach 2 (represented in Figure 3.5). There are several ways to perform a normalization (see Juszczak et al. [21]), one of which is using the formula

$$Y_t = \frac{X_t - mean(X_t)}{sd(X_t)} \tag{3.1}$$

where Y_t denotes the normalized time series, X_t is the original time series and sd stands for the standard deviation. This was the one that performed better (when comparing the final results of the clustering and the validating indexes), so this was the normalization method that we end up choosing.

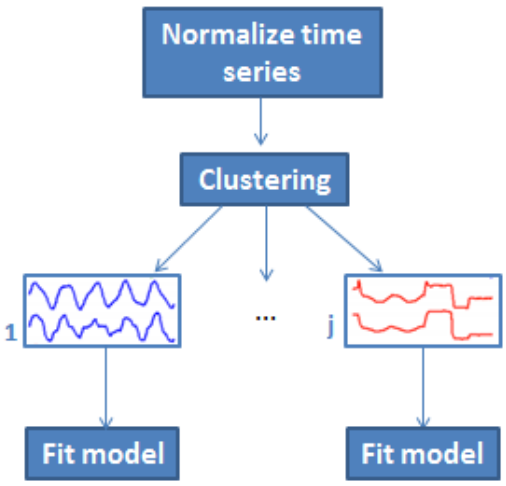

Figure 3.5: Approach 2 for clustering and fitting models for j clusters.

The following Subsections present in detail the results of the clustering for approach 2, including an explanation of the choices of agglomeration method of the hierarchical method and the number of clusters.

3.2.1 Hierarchical Clustering Algorithm

The time series will be clustered by applying hierarchical clustering algorithm with the periodogram distance, which is based on the periodograms of the time series and hence in the frequency domain of the time series, which goes in favour of our aim of grouping based on the consumer behaviour.

Hierarchical algorithm requires choosing a linkage method. We clustered the 25 given time series using the linkage methods average, complete and single, so that we have 3 final dendrograms for each linkage method. In order to find out which is the best linkage method, we choose the one having the maximum Dunn index for every partition. According to Table 3.1, having partitions of the clusters that vary from 2 clusters until 7 clusters, we can conclude that the average linkage method has the maximum Dunn index for every number of clusters, thus we choose this linkage method to perform the clustering.

Agglomeration Method	**2**	**3**	**4**	**5**	**6**	**7**
Average	0.43	0.43	0.43	0.43	0.45	0.46
Complete	0.43	0.35	0.43	0.36	0.39	0.40
Single	0.31	0.43	0.38	0.38	0.43	0.41

Table 3.1: Dunn index for each agglomeration method and varying the number of clusters from 1 to 7.

The dendrogram that results from clustering the 25 time series using the hierarchical algorithm with the average linkage method and distance matrix returned by the periodogram distance is presented in Figure 3.6.

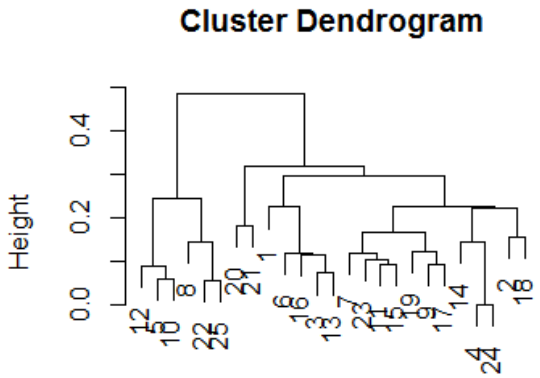

Figure 3.6: Dendrogram using hierarchical clustering with average linkage method and the periodogram distance.

3.2.2 Choosing the Best Number of Clusters

The height at which we will cut the dendrogram will control the number of clusters returned. We want to choose a partition that has the optimal number of clusters. This choice is based on the majority vote of the best number of clusters returned by four validation indexes: Dunn, Entropy, Silhouette and Gamma index. For each index, a plot of the number of clusters *versus* the value of the index was created (Figures 3.7, 3.8, 3.9, 3.10). Table 3.2 shows the best number of clusters returned by each index.

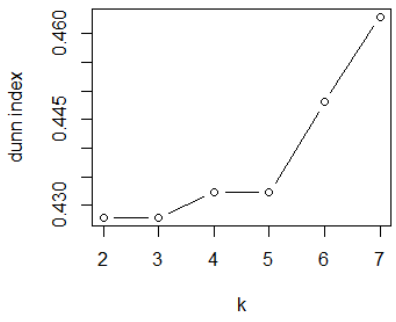

Figure 3.7: Number of clusters *versus* the value of Dunn index.

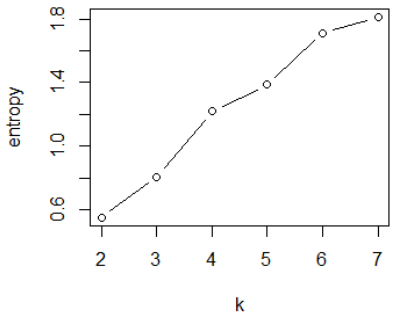

Figure 3.8: Number of clusters *versus* the value of Entropy.

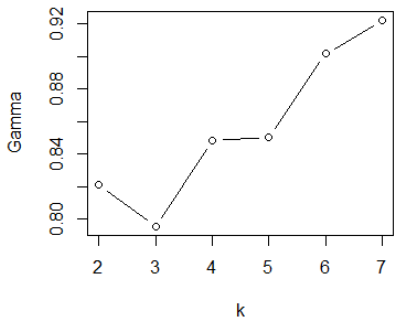

Figure 3.9: Number of clusters *versus* the value of Gamma index.

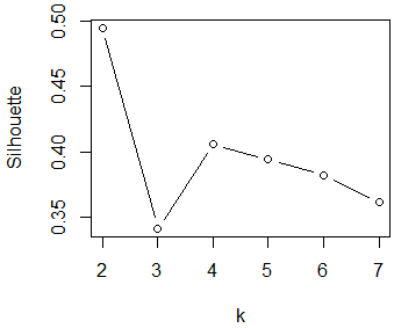

Figure 3.10: Number of clusters *versus* the value of Silhouette index.

	Dunn	**Entropy**	**Silhouette**	**Gamma**
Best number of clusters	7	2	2	7

Table 3.2: Best number of clusters according to four indexes using hierarchical clustering with the periodogram distance and average linkage method. Indexes computed from 2 to 7 number of clusters.

Both 2 and 7 were a good number of clusters. Since we have 25 time series to partition, we believe that a partition in 2 groups would be a very rough partition, so we will not choose 2 has to be the number of clusters. On the other hand, intuitively, we believe that 7 is already too many clusters to partition only 25 time series. In fact, having the partition of 6 clusters (Figure 3.11), partitioning in 7 clusters will only put one time series (time series 1) in a new cluster. So, considering the fact that the values of the indexes for 6 and 7 clusters do not differ too much, we decided to partition the data in 6 clusters, which leads to partitioning the dendrogram as in Figure 3.11.

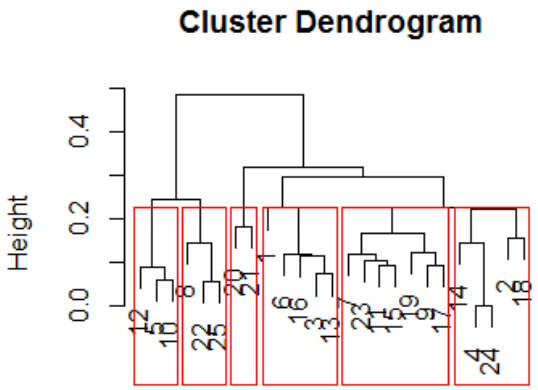

Figure 3.11: Partition of the 25 time series in 6 clusters.

3.2.3 Result of the Clustering

Let the labels of the resulting clusters be C_1, C_2, C_3, C_4, C_5 and C_6, where $C_1 = \{1, 3, 6, 13, 16\}$, $C_2 = \{2, 4, 14, 18, 24\}$, $C_3 = \{5, 10, 12\}$, $C_4 = \{7, 9, 11, 15, 17, 19, 23\}$, $C_5 = \{8, 22, 25\}$ and $C_6 = \{20, 21\}$.

For a better understanding of the result, this Subsection is aimed at giving a detailed exploratory analysis of the groups. See confidential annex

Having set the partition of the 25 time series, the next step is to fit a model for the representative time series of each cluster.

3.3 Modelling using GAM

Our aim is to fit a Generalized Additive Model in order to model and forecast electricity load of the time series within one cluster. The advantage of this model is that it allows the inclusion of external variables in the model and it is able to capture non-linear effects of the variables.

We will describe in detail the procedure for modelling and forecasting time series in cluster 6. For the remaining clusters, modelling and forecasting follow the same reasoning.

3.3.1 Representative Time Series of the Cluster

For cluster C_6, which contains time series 20 and 21, the input and output of our procedure is the following:

- **input:** Timestamp, time series 20, time series 21, 7 distinct weather variables from the location of time series 20, 7 distinct weather variables from the location of time series 21;
- **output:** One day ahead forecast of electricity consumption of the representative time series of the cluster, for every 15 minutes.

Given the normalized time series within cluster C_6 and without performing further transformation to the data, we started by getting the representative time series. The model will be fitted to the obtained time series. The exploratory analysis performed in Section 3.3.5 should be helpful in understanding which should be the underlying variables for the model. A quick brief on this analysis is:

- The representative time series data is stationary;
- There are two peaks of consumption within one day: one during mornings and another one during evenings;
- Work days have similar consumption mean, with greater mean value than weekend days;
- There is not a significant difference in the mean between months;

3.3.2 Explanatory Variables of the GAM

The next steps are aimed at fitting a GAM to the data. For that, we need to identify which are the best variables that should be included in the model. According to the exploratory analysis, we consider that the following explanatory variables should be included:

- Day type: categorical variable that represents the day type. We have one factor for each week day; also, it is important to set a factor for public holidays, for which consumption may be distinct from the remaining types of days. So, the factors of this variable are then 1 for Mondays, 2 for Tuesdays, 3 for Wednesdays, 4 for Thursdays, 5 for Fridays, 6 for Saturdays, 7 for Sundays, and 8 for public holidays;

- Time of day: categorical variable that represents the current time within the day (measured in quarter-hourly time steps). The total number of observation within one day is 96, because it is the result from multiplying 24 hours by 4, where 4 is the number of 15 minutes periods within one hour. So, the factors go from 0, representing hour 00 and minute 00 of the day, until 95, representing hour 23 and minute 45 of the day;

- Time of year: categorical variable that represents the current day and month within the year. The factors go from 0, representing the 1st of January, until 365, representing the 31st of December;

- Lags of electricity consumption. In order to find out which is the right lag of consumption, we make use of the Cross Correlation Function (CCF), which is helpful to find out, given two time series, y_t and x_t, whether the series y_t may be related to past lags of the x-series by identifying lags of the x-variable that might be useful predictors of y_t. The sample CCF is defined as the set of sample correlations between x_{t+h} and y_t for $h = 0, \pm1, \pm2, \pm3$, and so on. For data pairs $(x_1, y_1), ..., (x_T, y_T)$, an estimate of the sample CCF is given by the formula (Box and Reinsel [22]):

$$r_{xy}(h) = \frac{c_{xy}(h)}{s_x s_y}, h = 0, \pm1, \pm2, \pm3, ...$$

where $c_{xy}(h)$ is an estimate of the lag h cross-covariance, given by

$$c_{xy}(h) = \begin{cases} \frac{1}{T}\sum_{t=1}^{T-h}(x_t - \bar{x})(y_{t+h} - \bar{y}) & , h = 0, 1, 2, ... \\ \frac{1}{T}\sum_{t=1}^{T-h}(x_t - \bar{x})(y_{t-h} - \bar{y}) & , h = 0, -1, -2, ... \end{cases}$$

and s_x, s_y, $\bar{x}$ and $\bar{y}$ are the sample standard deviations and the sample means of the series x and y, respectively. Therefore, if we compute the sample CCF of the electricity consumption with it self, the lag with greater sample CCF should be the one to be included in the model. Figure 3.12 represents the sample CCF of the load with it self.

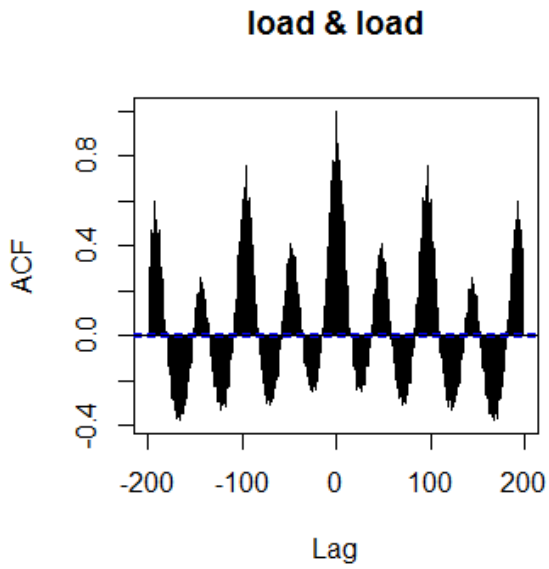

Figure 3.12: Sample CCF of variable load with it self.

Recall that one day of measurements of electricity consumption has 96 observations. In order to predict the load for one day ahead, we cannot consider lags under 96. So, lag 96 is the one with greater sample CCF under this condition, and this is the one that we should include in the model.

- Lags of weather variables. Notice that time series within the cluster are from different locations and we are fitting a model for the aggregated model, which means that we need to decide from which location should we choose the weather variables. Our strategy to solve this problem is the following: we build one GAM using weather variables of the location of time series 20, another GAM using weather variables of the location of time series 21 and the final GAM will be an ensemble, i.e., the mean of the two last generalized additive models (for clusters that have more than 2 time series, we choose two time series at random to perform the ensemble).

Recall that we are given seven weather variables: radiance, atmospheric pressure, temperature, wind direction, wind speed, wind U and wind V. However, some of them might be correlated with each other, hence, we shall detect which ones are correlated with each other and only include in the model not correlated variables. We start by considering the weather variables of time series 20. Figures 3.13, 3.14, 3.15, 3.16, 3.17 and 3.18 show the sample CCF between the temperature and the remaining weather variables associated with time series 20.

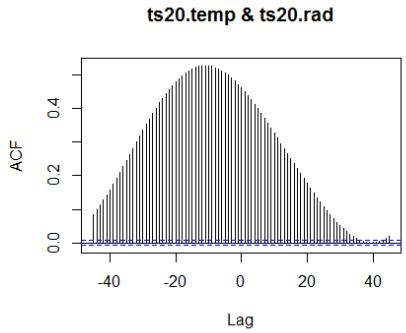

Figure 3.13: Sample CCF between temperature and radiance.

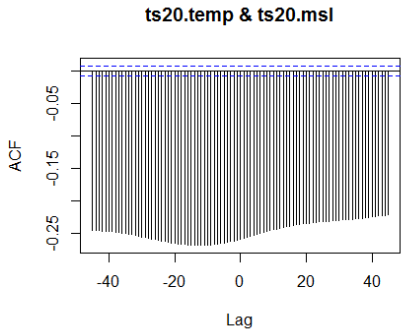

Figure 3.14: Sample CCF between temperature and atmospheric pressure.

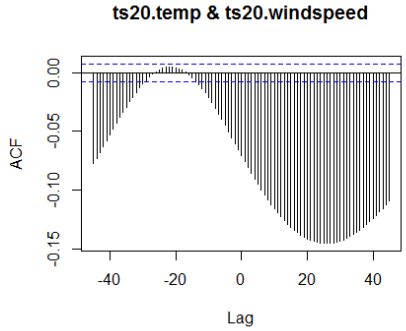

Figure 3.15: Sample CCF between temperature and wind speed.

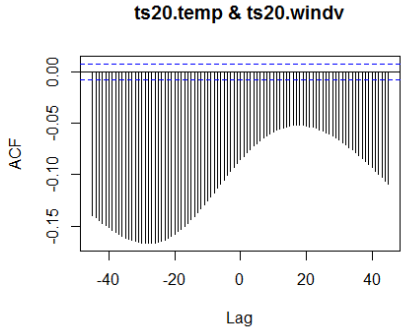

Figure 3.16: Sample CCF between temperature and wind V.

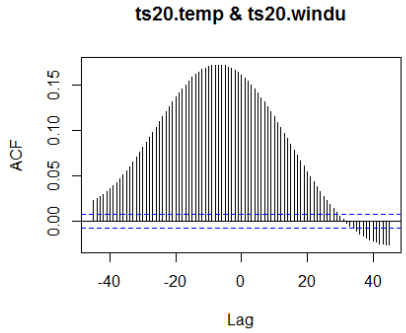

Figure 3.17: Sample CCF between temperature and wind U.

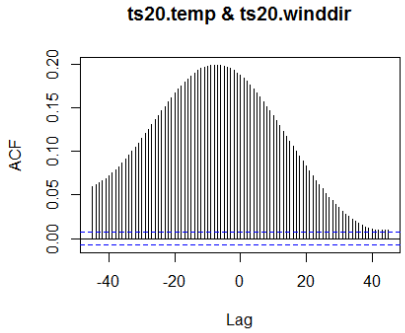

Figure 3.18: Sample CCF between temperature and wind direction.

By analysing the sample CCF between temperature and the remaining weather variables, we can conclude that the temperature is mostly related with radiance, atmospheric pressure and wind direction, since we can find values of correlation greater than 0.15 for these variables. On the other hand, temperature is less related with wind U, wind V and wind speed, having the higher values of the sample CCF around 0.15. So, if we include temperature in the model, it will make sense to include as well one of these wind variables that are not correlated with the temperature. Figures 3.19, 3.20 and 3.21 are aimed at analysing the correlation between all wind variables.

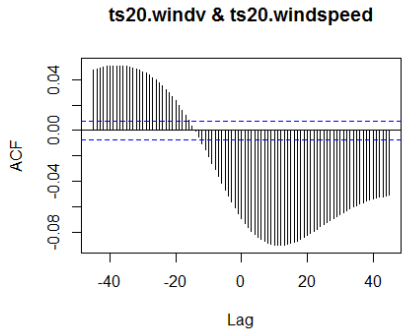

Figure 3.19: Sample CCF between wind V and wind speed.

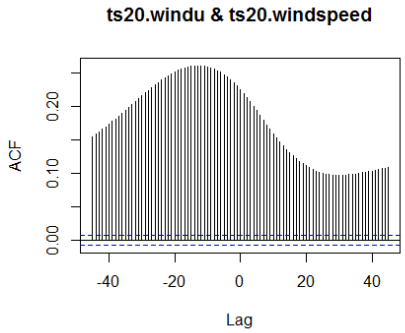

Figure 3.20: Sample CCF between wind U and wind speed.

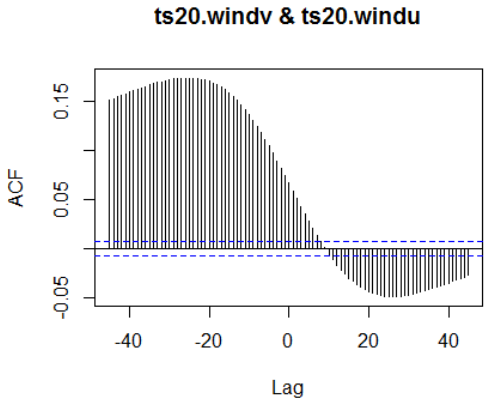

Figure 3.21: Sample CCF between wind V and wind U.

By analysing Figure 3.20 we conclude that wind U and wind speed are highly correlated, with the maximum sample CCF having value above 0.25. Therefore it makes sense to include either wind U or wind speed in the model, but not both. Therefore, we choose to include in the model the variable wind speed, since it is related with wind U and wind U is related with wind V (sample CCF greater than 0.15 in Figure 3.21).

So, the first GAM will be built with variables temperature and wind speed from the location of time series 20. A similar study of the weather variables from the location of time series 21 resulted in the same choice of variables for the second GAM.

It remains to see, from the weather variables chosen, which are the lags that should be included in the model. The lags chosen are the ones that represent the greater sample CCF between the weather variables and the load.

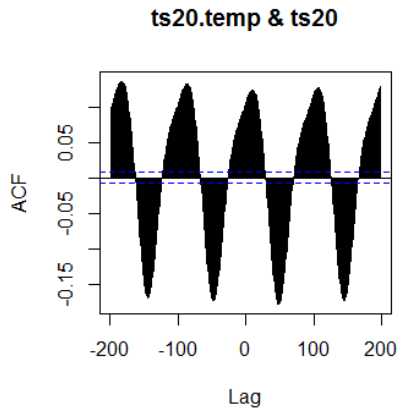

Figure 3.22: Sample CCF between temperature and load for model 1.

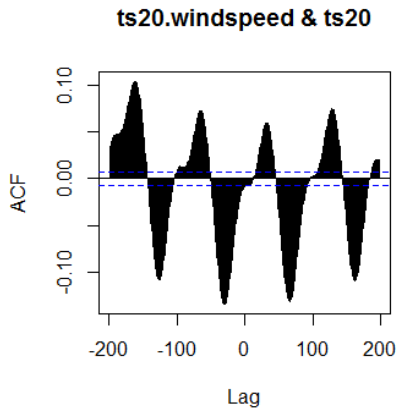

Figure 3.23: Sample CCF between wind speed and load for model 1.

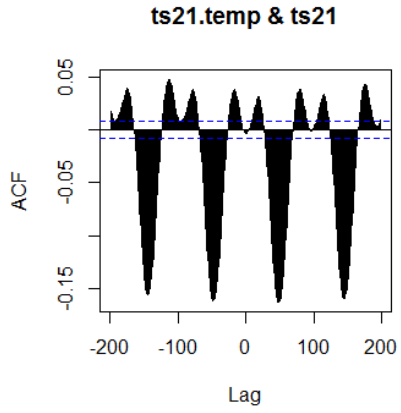

Figure 3.24: Sample CCF between temperature and load for model 2.

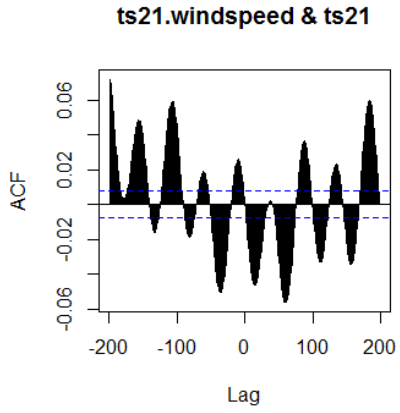

Figure 3.25: Sample CCF between wind speed and load for model 2.

For the first GAM model, according to Figures 3.22 and 3.23, the temperature lag should be 48, and the wind speed lag should be 48 as well. For the second GAM model, according to Figures 3.24 and 3.25, the temperature lag should be 48, whereas the wind speed lag should be 198.

To sum up, since cluster 6 is composed by 2 time series from different locations, we build two Generalized Additive Models, with covariates $x1_t$ and $x2_t$, respectively, given by:

$$x1_t = \{daytype_t, timeday_t, timeyear_t, y_{t-96}, ts20.temperature_{t-48}, ts20.windspeed_{t-48}\}$$

$$x2_t = \{daytype_t, timeday_t, timeyear_t, y_{t-96}, ts21.temperature_{t-48}, ts21.windspeed_{t-198}\}$$

where y_t denotes the values of the aggregated time series of electricity consumption at time t.

3.3.3 Train and Test Set

We split the aggregated time series y_t (which is the response variable) and the covariates $x1_t$ and $x2_t$ in a train set (used for fitting the model) and in a test set (used for compare forecast with true values).

Our aim is to create two models that fit the aggregated data until the 9th of January of 2017 and to predict the load curve for the 10th of January of 2017, which is a Tuesday. So the train set will be the observations from the 5th of May of 2015 to the 9th of January of 2017 and the test set will be from the 10th of January of 2017 until the 19th of March of 2017.

3.3.4 Modelling

We fit the following two generalized additive models for the electricity load:

$$\begin{aligned} \textbf{Model 1: } y_t = {} & \beta^{intercept} + f(trend_t) + f(y_{t-96}) + f(ts20.temperature_{t-48}) \\ & + f(ts20.windspeed_{t-48}) + f(timeyear_t) + \sum_{l=1}^{8} I_{\{daytype_t=l\}} f(timeday_t) \end{aligned} \tag{3.2}$$

$$\begin{aligned} \textbf{Model 2: } y_t = {} & \beta^{intercept} + f(trend_t) + f(y_{t-96}) + f(ts21.temperature_{t-48}) \\ & + f(ts21.windspeed_{t-198}) + f(timeyear_t) + \sum_{l=1}^{8} I_{\{daytype_t=l\}} f(timeday_t) \end{aligned} \tag{3.3}$$

where:

- $\beta^{intercept}$ models the base load and $f(trend_t)$ captures non-linear trends;
- $f(y_{t-96})$ takes into account the electricity load of the previous day;
- $daytype_t$ and $f(timeday_t)$ capture the day-type specific effects of the time of the day;
- $f(ts20.temperature_{t-48})$ and $f(ts20.windspeed_{t-48})$ take into account, respectively, the temperature and the wind speed of the previous half day from the location of the client related to time series 20;
- $f(ts21.temperature_{t-48})$ and $f(ts21.windspeed_{t-198})$ take into account, respectively, the temperature of the previous half day and the wind speed of the previous two days from the location of the client related to time series 21, and
- $f(timeyear_t)$ represents yearly cycles.

Thin plate regression splines is the basis used by default to represent the smooth functions f, because these are the optimal smoother of any given basis dimension. In fact, we tried other basis, but the change in basis has made very little difference to the fit, which means that the model does not depended very strongly on details such as the exact choice of basis.

Another choice in the previous two models is the choice of the dimension, k, of the basis used to represent smooth terms, where the default, $k = 10$, was used. The choice of basis dimensions amounts to setting the maximum possible degrees of freedom allowed for each model term. The actual effective degrees of freedom, for each term, will usually be estimated from the data, by GCV, but the upper limit on this estimate is $k - 1$: the basis dimension, less one degree of freedom due to the identifiability constraint on each smooth term. We did not change the default value in order to avoid overfitting.

The final model, which we call ensemble model, will correspond to the mean of the fitted models 1 and 2 at each observation.

3.3.5 Analysis of the Residuals

Having the ensemble model, the next step aims at analysing the residuals, namely, its stationarity and normality.

KPSS test, which tests a null hypothesis that an observable time series is stationary, returns a p-value greater than 0.1. We reject the null hypothesis significance levels greater than the p-value, so, for the usual significance levels (1%, 5% and 10$\%$), we do not reject the null hypothesis, meaning that the residuals are stationary.

However, by analysis of the histogram (Figure 3.26) and QQ-plot (Figure 3.27) of the aggregated time series, the aggregated time series does not seem to follow a Normal distribution. This means that we are not able to produce forecast intervals for the prediction. One alternative approach to provide a maximum and minimum value for the electricity consumption will be stated in Section 3.3.7.

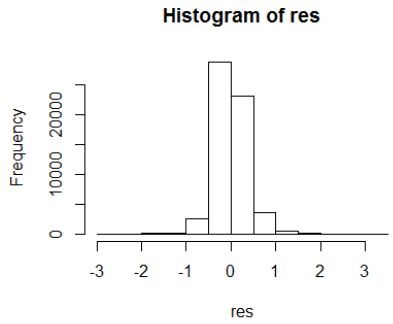

Figure 3.26: Histogram of the residuals of the ensemble model.

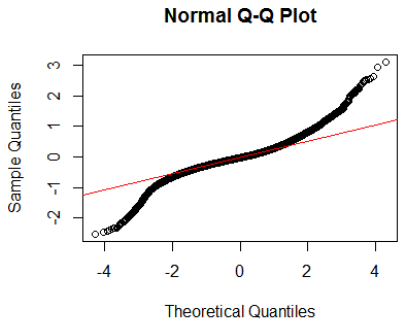

Figure 3.27: QQ-plot of the residuals of the ensemble model.

Figures 3.28 is the graphic of the residuals *versus* the linear predictor. The points appear randomly around zero, which indicates that the residuals are uncorrelated.

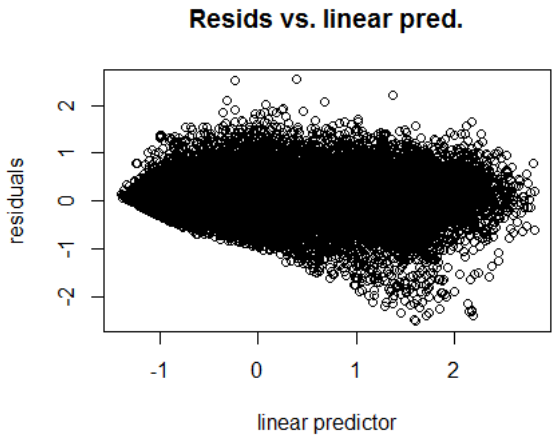

Figure 3.28: Linear predictor *versus* residuals.

3.3.6 Forecasting

The one day prediction for the aggregated normalized time series of cluster 6 is presented in Figure 3.29. Green lines represent the test values, whereas blue lines represent the forecast.

Notice that we are dealing with normalized values. The next step is to transform the final predictor vector into its original scale using the formula:

$$Z_t = Y_t * sd(X_t) + mean(X_t) \tag{3.4}$$

where Y_t denotes the normalized predictor vector, X_t is the non-normalized representative time series of the cluster and Z_t denotes the non-normalized predictor vector. Figure 3.30 compares the forecast with the true values of the load in its aggregated original scale.

Figure 3.29: Confidential annex: Forecast the normalized aggregated time series of cluster 6.

Figure 3.30: Confidential annex: Forecast the non-normalized aggregated time series of cluster 6.

Since we have the true aggregated values for the 10th of January of 2017, we intended to analyse the performance of the method, by comparing the forecast of the non-normalized aggregated time series of cluster 6 with the true aggregated values. We computed the MAPE and the result for this forecast was 16.6$\%$.

Note that all Figures that we referred until yet are composed by curves that represent aggregated values. Suppose now that a time series, ts, is classified into these cluster. Then, the non-normalized predictor vector is obtained by performing

$$Z_t = Y_t * sd(ts_t) + mean(ts_t). \tag{3.5}$$

3.3.7 Intervals for Consumption Forecast

In Section 3.3.5 we noticed that the residuals of the model were not normal, which means that we cannot make use of the normality to create prediction intervals. Our way of contour this problem was to fit, not only an ensemble model for the aggregated time series using the median, but also two more ensemble models: the first one will fit the aggregated time series where the function using to perform the aggregation is the minimum at each observation, and the second one will fit the aggregated time series where the function using to perform the aggregation is the maximum at each observation. The forecast of these two new ensemble models will give an idea of the maximum and minimum values of consumption.

The one day prediction for the aggregated normalized time series of cluster 6 using the median, minimum and maximum is presented in Figure 3.32. Green lines represent the test values, whereas red lines represent the predictor vector, with the dash lines representing the aggregated prediction using the minimum and the maximum function and the thick red line representing the aggregated prediction using the median.

Figure 3.31: Confidential annex: Forecast the aggregated time series using median, minimum and maximum at each observation of cluster 6.

Figures 3.32, 3.33, 3.34 and 3.35 show forecasts for cluster 6 for four different types of days. Similar figures were produced for the remaining clusters and these are stated at A.

Figure 3.32: Confidential annex: Forecast of cluster C_6 for a typical tuesday in winter.

Figure 3.33: Confidential annex: Forecast of cluster C_6 for a public holiday.

Figure 3.34: Confidential annex: Forecast of cluster C_6 for a typical Saturday on winter.

Figure 3.35: Forecast of cluster C_6 for a typical Tuesday on summer.

Notice that these intervals of consumption are not prediction intervals, since they have no associated confidence. In the next Section, we apply a method to obtain prediction intervals without the need of satisfying a normality condition.

3.3.8 Prediction Intervals using Bootstrap

The method of bootstrapping will be used to build prediction intervals. This method does not assume any distribution of the variables, which means that it is suitable for what we aim. It consists in taking samples (with replacement) from the observations, what we call as bootstrap samples, and for each sample we obtain the prediction vector. This way, we can get a 95$\%$ bootstrap percentile prediction interval from the samples (for further details on this method, see, for example, Vinod and de Lacalle, 2009).

The result of the bootstrap is presented in A. Due to the fact that the amplitude of the prediction interval is very small, in the sense that, when plotted, it is nearly indistinguishable from the one day ahead consumption forecast, we left this method out.

3.3.9 General Case

For the remaining clusters, as mentioned before, the procedure for fitting an ensemble GAM is the same. The only step that must be taking care when dealing with different clusters is the choice of variables of the model. Generally, let $ts1$ and $ts2$ be the two randomly chosen time series whose weather variables will constitute GAM 1 and 2, respectively. For the construction of the two GAM 1 and 2 we have the explanatory variables, respectively

$$x1_t = \{daytype_t, timeday_t, timeyear_t, y_{t-\alpha}, ts1.temperature_{t-\beta_1}, ts1.windspeed_{t-\gamma_1}\}$$

$$x2_t = \{daytype_t, timeday_t, timeyear_t, y_{t-\alpha}, ts2.temperature_{t-\beta_2}, ts21.windspeed_{t-\gamma_2}\}$$

where α, β_1, β_2, γ_1 and γ_2 need to be selected based on the CCF function.

Following the procedure for fitting an ensemble GAM for the remaining clusters, we computed predictions for the 10th of January of 2017 (tuesday), the 1st of January of 2017 (public holiday), the 6th of January of 2017 (saturday) and finally for the 30th of August of 2016 (summer holidays). The MAPE for each cluster and day is presented in Table 3.3.

	10 JAN 2017	**1 JAN 2017**	**6 JAN 2017**	**30 AUG 2016**
C1	11.36	288.35	18.71	5.57
C2	5.27	1.63	12.67	21.62
C3	8.90	9.02	8.99	3.76
C4	10.40	7.60	8.74	7.93
C5	4.86	6.86	4.70	3.51
C6	16.6	23.4	19.98	15.71

Table 3.3: MAPE ($\%$) for the aggregated function of each cluster using the function median.

For cluster 1, we can see that there is an anomalous MAPE in the 1st of January. This may be due to the fact of the consumption during this public holiday being completely different from the remaining time of year, which makes it hard to predict. For the remaining cases, we can see that the MAPE varies between around 4$\%$ and 23$\%$.

3.4 Modelling using SARIMA

In order to compare the performance of the Generalized Additive Model, we fit a seasonal ARIMA model to the representative time series of the cluster, that is, a $SARIMA(p,d,q)(P,D,Q)[m]$ model. Once again, we illustrate the model procedure for cluster 6 and forecast the load for a typical Tuesday on winter.

3.4.1 Parameters Identification

Since the representative time series of the cluster is stationary, there is no need to differentiate, so $d = D = 0$. The frequency of the time series is one day, that is, 96 observations, hence $m = 96$.

Therefore, it remains to determine p, q, P and Q, using the auxiliary graphs of the sample ACF (Figure 3.36) and PACF (Figure 3.37).

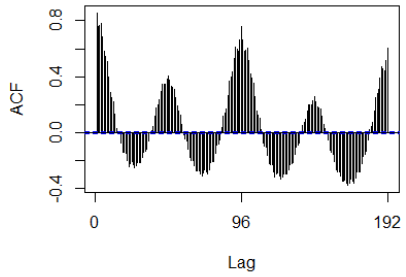

Figure 3.36: Sample ACF.

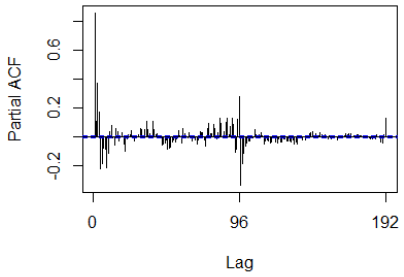

Figure 3.37: Sample PACF

By analysing Figures 3.36 and 3.37, our first proposal of model is $ARIMA(1,0,0)(1,0,0)[96]$, due to the peak at sample PACF at lag 96 and due to the sinusoidal pattern of the sample ACF. However, the incrementation of q was leading to a decreasing of the AICC score, which we want to be minimum. Hence, by analysis of the value of AICC, we reached the model $ARIMA(1,0,6)(1,0,0)[96]$.

3.4.2 Analysis of the Residuals

Having the model, the next step aims at analysing the residuals, namely, its stationarity and normality.

KPSS test, which tests a null hypothesis that a time series is stationary, returns a p-value greater than 0.1. We reject the null hypothesis significance levels greater than the p-value, so, for the usual significance levels (1%, 5% and 10%), we do not reject the null hypothesis, meaning that the residuals are stationary.

However, by analysis of the histogram (Figure 3.38) and QQ-plot (Figure 3.39) of the aggregated time series, the aggregated time series seems to be close to follow a Normal distribution but still presenting heavy tails, so, once again, we are not able to produce forecast intervals for the prediction.

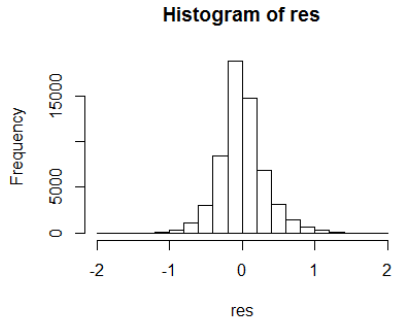

Figure 3.38: Histogram of the residuals of the SARIMA model.

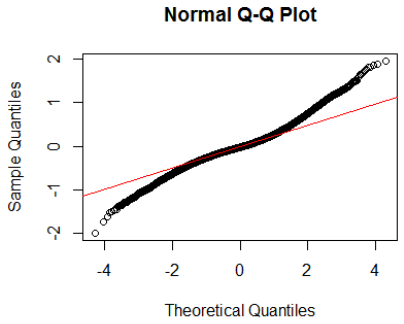

Figure 3.39: QQ-plot of the residuals of the SARIMA model.

3.4.3 Forecasting

The one day prediction for the aggregated normalized time series of cluster 6 is presented in Figure 3.40. Green lines represent the test values, whereas blue lines represent the forecast.

Figure 3.40: Confidential annex: Forecast the aggregated time series of cluster 6 using SARIMA.

3.4.4 Accuracy Analysis

After transforming the final predictor vector into its original scale, we computed the MAPE of the forecast. The result was 20.1%, which is higher value of MAPE than the one obtained for GAM, hence, in this case, the prediction is worst when fitting a SARIMA model.

3.4.5 General Case

For the remaining clusters, the procedure of fitting a SARIMA model is the same. The only step that must be taking care of and may be different from cluster to cluster is the choice the parameters of the model, which is done by analysing the sample ACF and PACF of each representative time series of the clusters.

Following the procedure for fitting a SARIMA model the remaining clusters, we computed predictions for the 10th of January of 2017 (Tuesday), the 1st of January of 2017 (public holiday), the 6th of January of 2017 (Saturday) and finally for the 30th of August of 2016 (summer holidays) by fitting a SARIMA model. The MAPE for each cluster and day is presented in Table 3.4. The cells with text coloured in green indicate that the forecast returns lower MAPE compared with GAM and, hence, it is better to fit the SARIMA model in those cases.

	10 JAN 2017	**1 JAN 2017**	**6 JAN 2017**	**30 AUG 2016**
C1	24.50	**57.81**	70.74	29.24
C3	21.31	29.25	29.11	34.36
C4	22.99	30.30	34.17	28.11
C5	26.59	19.32	33.60	30.16
C6	20.12	25.61	**18.81**	20.29

Table 3.4: MAPE (%) for the aggregated function of each cluster using the function median and a SARIMA model.

Using a SARIMA model, the values of MAPE go from approximately 18% until 35%, and we can see an anomalous value of 70% during a saturday for cluster 1. The value of MAPE is lower for only 2 cases out of 20 when comparing to the Generalized Additive Model. This means that, overall, fitting a SARIMA model is not better than fitting a Generalized Additive Model.

Chapter 4

Conclusions

This Chapter is dedicated to state the achievements (Section 4.1) obtained thought this study plus a suggestion of directions for future work (Section 4.2).

4.1 Achievements

In this study, we were focused in forecasting electricity consumption, so we were given a set of 25 time series that contained measures of consumption of 25 different clients through almost 2 years.

It is impractical to create models and to forecast consumption for each of the time series given, so it was necessary to start by performing a clustering method, in order to group clients by similarity of consumption. The chosen method was hierarchical clustering and two approaches were analysed: either we could apply the clustering method to raw data or to normalized data. We concluded that, when the data were first normalized, then we obtained partitions that could best reflect the consumption patterns of the clients (hourly, daily or monthly behaviours); on the other hand, in the presence of raw data, the clustering would be focused mainly on scale. Since we were more interested in creating models posteriorly that would translate trends and seasonalities of the data, we conclude that it is necessary to perform a normalization. The distance that we applied when performing the clustering was based in the periodograms of the normalized time series, because this was the most appropriated one that goes along with our objective of extracting and comparing the patterns of time series. The best number of clusters was determined by analysis of several validation indexes.

Having the right partition of the 25 time series according to their similarity, we aimed at building a model that could represent the consumption of each cluster and comparing the forecast accuracy of two models: Generalized Additive Models and SARIMA models. The GAM has the advantage of including external variables and capturing non-linear effects of the variables. In fact, before the modelling phase, having performed a proper exploratory analysis to each cluster, such as plotting daily or monthly patterns, can help us understand which underlying variables should be included in the model. For instance, we could conclude from this analysis that summer months and winter months have different consumption mean. Therefore, we were also given weather variables so that we could studied the correlation between

these and the consumption. However, notice that each of the 25 clients are from different locations and recall that the model is being fitted to a time series that represents the consumption of the whole cluster, which means that we had to decide which locations should be included in the model. So, for the GAM, we created an ensemble model: given a cluster of clients, we chose the locations of two clients at random (two because of computational costs, but this number could be extended) and we created one model with weather variables from one client and another model with weather variables from the other. The ensemble is defined to be the mean of the two fitted models. On the other hand, SARIMA models include only lagged consumption and lagged errors, so the problem of having different locations does not arise. The right lags are chosen by analysis of the ACF and the PACF of the representative time series of the cluster. Finally, by comparing the forecast accuracy of the GAM and the SARIMA models, we could understand that, in general, the GAM performs better.

So, to summarize, we conclude that the normalization of time series followed by hierarchical clustering with the periodogram distance is the best way to cluster, in order to get partitions of clients based on their consumption behaviour. Furthermore, fitting a Generalized Additive Model to the representative time series of each cluster returned the best one-day ahead forecasts. Along with the point forecast, we were able to as well indicate maximum and minimum limits for the forecast, which gives us an idea of intervals of consumption.

4.2 Future Work

The proposed methods in this thesis were applied to 25 time series. The suggestion of future work is to apply the same methods to a larger set of time series. Assume that we have a set S_N that contains N load curves, with $N >> 25$. Then we apply the clustering methods to the set S_N and we fit a GAM to each cluster. Now suppose that we have a new load curve that was not in the set S_N. Then, we could apply classification methods, such as the K-Nearest Neighbours, to classify the new time series in one of the clusters, identified previously, and forecast its consumption by making use of the aggregated model of that cluster. In other words, we suggest not to apply the clustering methods to the whole customer portfolio, only part of it, and classify the remaining customers into the right clusters, based on classifying methods.

Bibliography

[1] R. Hyndman, X. Wang, and K. Smith. Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery*, 13:335–364, Dec. 2006. DOI:10.1007/s10618-005-0039-x.

[2] T. Rasanen, D. Voukantsis, H. Niska, K. Karatzas, and M. Kolehmainen. Data-based method for creating electricity use load profiles using large amount of customer-specific hourly measured electricity use data. *Energy*, 87:3538–3545, 2010. DOI:10.1016/j.apenergy.2010.05.015.

[3] Y. Goude and J. Cugliari. Disaggregated electricity forecasting using wavelet-based clustering of individual consumers. In *Energy Conference (ENERGYCON)*. IEEE International, July 2016. DOI: 10.1109/ENERGYCON.2016.7514087.

[4] G. Chicco. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy*, 42:68–80, Jan. 2012. DOI:10.1016/j.energy.2011.12.031.

[5] W. Kim. Parallel clustering algorithms: Survey. *CSC 8530 Parallel Algorithms*, 2009.

[6] S. J. Huang and K. R. Shih. Short-term load forecasting via arma model identification including non-gaussian process considerations. *IEEE Transactions on Power Systems*, 18:673 – 679, June 2003. DOI: 10.1109/TPWRS.2003.811010.

[7] J. W. Taylor. Short-term load forecasting with exponentially weighted methods. *IEEE Transactions on Power Systems*, 27:458 – 464, Mar. 2012. DOI: 10.1109/TPWRS.2011.2161780.

[8] V. Bianco, O. Manca, and S. Nardini. Electricity consumption forecasting in italy using linear regression models. *Energy*, 34:1413–1421, July 2009. http://DOI.org/10.1016/j.energy.2009.06.034.

[9] A. B. Yannig Goude, Pascal Pompey and M. Sinn. Adaptive learning of smoothing functions: Application to electricity load forecasting. In *Advances in Neural Information Processing Systems 25*, pages 2519–2527. NIPS, January 2012.

[10] W. Wei. *Time Series Analysis: Univariate and Multivariate Methods*. Pearson, 2^{nd} edition, 2006. ISBN 0-321-32216-9.

[11] G. Žitković. Introduction to stochastic processes - lecture notes, 2010.

[12] P. J. Brockwell and R. A. Davis. *Introduction to Time Series and Forecasting*. Springer, 2^{nd} edition, 2002. ISBN 0-387-95351-5.

[13] R. J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. Otexts, 2^{nd} edition, 2014.

[14] A. Pacheco. *Notas de Séries Temporais*. Instituto Superior Técnico, March 2001.

[15] S. N. Wood. *Generalized Additive Models: an introducing to R*. Chapman and Hall/CRC, 2006. ISBN 9781584884743.

[16] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*. Springer, 2^{nd} edition, 2015. ISBN 978-1-4614-7137-0.

[17] P. Montero and J. A. Vilar. Tsclust: An R package for time series clustering. *Journal of Statistical Software*, 62, 2014.

[18] J. Caiado, N. Crato, and D. Peña. A periodogram-based metric for time series classification. *Elsevier*, 50:2668 – 2684, Sept. 2014. DOI:10.1016/j.csda.2005.04.012.

[19] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs. NbClust: An R package for determining the relevant number of clusters in a data set. *Journal of Statistical Software*, 61, Oct. 2014.

[20] F. T. Denton. Adjustment of monthly or quarterly series to annual totals: An approach based on quadratic minimization. *Jornal of the American Statistical Association*, 66:99 – 102, Mar. 1971.

[21] P. Juszczak, D. Tax, and R. Duin. Feature scaling in support vector data description. DOI: 10.1.1.100.2524.

[22] G. M. J. Box, G. E. P. and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice Hall, 3^{rd} edition.

[23] H. D. Vinod and J. L. de Lacalle. Maximum entropy bootstrap for time series: The meboot r package. *Journal of Statistical Software*, 29, 2009.

Appendix A

Confidential Data