Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

A hierarchical model to electricity load forecasting

DEGREE FINAL WORK

Degree in Computer Engineering

Author: Stéphane Díaz-Alejo León
Tutor: Lenin Guillermo Lemus Zúñiga
Cotutor: Eva Onaindía de la Rivaherrera
Experimental Director: Diego Aineto García

Course 2020-2021

Acknowledgements

First of all, I would like to thank the cooperation shown by the city council of Llíria in providing us with their electricity consumption data. Secondly, I would like to thank the Information & Communications Technologies vs Climate Change (ICTvsCC) team of the Polytechnic University of Valencia, with whom I have collaborated this year through a collaboration grant from the ministry, for offering me the idea of this project. Last but not least, I would like to acknowledge the great effort that my tutors, Lenin Guillermo Lemus and Eva Onaindia, and my experimental director, Diego Aineto, have dedicated to the development of this project at all levels, but especially in the areas of programming and writing.

Resum

Els models de predicció de consum elèctric han cridat l'atenció en l'última dècada a causa de que són de gran ajuda per a la modernització de la xarxa elèctrica i, en conseqüència, per al medi ambient. En particular, els models que utilitzen informació jeràrquica han demostrat produir més interès darrerament En aquest projecte, proposem construir un model jeràrquic de previsió de consum elèctric per predir el consum de l'endemà. Atès que les dades de consum elèctric es distribueixen en una zona geogràfica relativament petita es proposa utilitzar una agrupació natural de les dades i deduir automàticament una jerarquia basada en la similitud el perfil de consum dels diferents punts de mesura. L'objectiu principal d'aquest projecte és crear un model precís que ajudi en la presa de decisions de les comunitats locals per reduir l'impacte mediambiental i disminuir els costos econòmics. Per això s'han utilitzat les dades de consum horari dels edificis públics gestionats per l'ajuntament de Llíria.

Paraules clau: model predictiu, agrupació jeràrquica, consum elèctric, machine learning

Resumen

Los modelos de predicción de consumo eléctrico han llamado la atención en la última década debido a que son de gran ayuda para la modernización de la red eléctrica y, en consecuencia, para el medio ambiente. En particular, los modelos que utilizan información jerárquica han demostrado producir más interés últimamente. En este proyecto, proponemos construir un modelo jerárquico de previsión de consumo eléctrico para predecir el consumo del día siguiente. Dado que los datos de consumo eléctrico se distribuyen en una zona geográfica relativamente pequeña se propone utilizar una agrupación natural de los datos y deducir automáticamente una jerarquía basada en la similitud del perfil de consumo de los diferentes puntos de medida. El objetivo principal de este proyecto es crear un modelo preciso que ayude en la toma de decisiones de las comunidades locales para reducir el impacto medioambiental y disminuir los costes económicos. Para ello se han utilizado los datos de consumo horario de los edificios públicos gestionados por el ayuntamiento de Llíria.

Palabras clave: modelo predictivo, agrupación jerárquica, consumo eléctrico, machine learning

Abstract

Load forecasting models have drawn attention over the last decade, due to the fact that are of great help to the modernization of the power grid, and consequently, to the environment. In particular, models that use hierarchical information have shown to produce more interest lately. In this project, we propose to build a hierarchical electricity consumption forecasting model to predict the next day's consumption. Since the electricity consumption data is distributed over a relatively small geographical area, it is proposed to use a natural clustering of the data and automatically derive a hierarchy based on the similarity of the consumption profile of the different measurement points. The main objective of this project is to create an accurate model that will help on the decision making of local communities to reduce the environmental impact and cut economical costs. For this purpose, the hourly consumption data of the public buildings managed by Llíria's town hall were used.

Key words: predictive model, clustering, electricity load, machine learning

Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

1.1 Motivation

Load forecasting refers to the prediction of the power/energy needed to meet the demand and supply equilibrium by power or energy-providing companies. Nowadays, load prediction is not only an issue of energy companies but also of entities and communities that wish for a reasonable energy consumption planning.

For that same reason, over the last decade, a large number of countries have dedicated increasing efforts to the modernization of the power grid. This endeavour is an enormous step to fight the climate change and will help to accomplish the Paris Agreement of 2015 and the seventh Sustainable Development Goal (Ensure access to affordable, reliable, sustainable and modern energy).

The modernization process in power grids has precipitated the deployment of smart meters, devices that record the consumption of electric power (typically at an hourly rate) and communicate it to some central system for monitoring and billing purposes. The staggering amounts of data generated by such devices has brought up a new challenge to the forecasting community, that now has access to high resolution datasets [11]. Furthermore, these datasets are hierarchical in nature as smart meters are grouped by electric utilities following geographical and structural factors.

It is of no surprise then that, currently, one of the most promising lines of research lies in exploiting this hierarchical structure as is evidenced by the last Global Electric Forecasting Competition's focus on hierarchical load forecasting [24]. Exploiting hierarchies is useful because at high levels of the hierarchy the data shows very few irregularities compared to lower levels which tend to be much noisier [24, 23].

The proposed approaches to hierarchical load forecasting consider datasets generated by large scale utilities where a hierarchy is already defined, usually following a spatial criteria. This kind of hierarchies are useful because they implicitly encode very valuable information. For instance, in a hierarchy based on a spatial criteria the sibling leaf nodes are smart meters that belong to the same region and, therefore, they share similar weather and luminosity conditions, which are factors known to impact the electricity consumption [6].

In this project, however, we focus our attention in small electric utilities where the hierarchy is unknown and spatial hierarchies cannot be exploited as the smart meters are too closely located. This setting occurs in some countries like Spain, where local communities are entitled to act as direct players in the electricity pool (known as "direct electricity consumer") and can also negotiate directly in the electricity market. In such cases, where a spatial hierarchy is not meaningful, we claim that a data-driven approach is a more appropriate solution as it enables to automatically infer a hierarchy grounded on other factors like the different consumption behaviour that meters exhibit.

To sum up, the problem in matter is to accurately forecast the hourly load consumption of different smarts meters and the aggregation of these points to obtain the total consumption of an energy consumer.

1.2 Objectives

The main objective of this work is to create an accurate load forecasting system that will help on the decision making of local communities to reduce the environmental impact and cut economical costs. In order to solve the problem in question, we will pursue the following two objectives:

- To construct a hierarchical structure that splits load series at diverse levels following a natural grouping of data. The hierarchy will provide comprehensive information on electricity consumption at different levels and will help identify critical points of the power grid. To achieve this objective we propose the subsequent tasks:
  - Preprocess the dataset by correcting anomalies and completing missing data.
  - Define the set of features that represents a smart meter.
  - Test a range of methods for establishing the data-driven hierarchy.
- To design and develop a load forecasting model that predicts the hourly load consumption of different smarts meters and the aggregation of these points to obtain the total consumption of an energy consumer. With this objective in mind, we have defined the ensuing tasks:
  - Design and build a predictor on each node of the hierarchy that takes into account the seasonality of the load consumption data and uses the hierarchical information.
  - Research and explore different methods and/or structures to combine and reconcile the lower level predictions with the higher ones in the hierarchy.
  - Compare the performance of the different configurations of hierarchy.

1.3 Organization of the document

This work is structured as follows. First, in Chapter 2, we will discuss the current state of the art in Load Forecasting. In section 2.1 we will introduce the concept of Load Forecasting and its characteristics, followed by section 2.2 which explains the different approaches to produce a load forecasting model. The chapter ends with section 2.3 that addresses the hierarchical forecasting models.

Chapter 3 discusses the details of a load forecasting problem applied to the small village of Llíria. First, in section 3.1 we formally introduce the steps to solve our particular problem. Second, in section 3.2, we spend a little time explaining the details of the day-ahead forecasting. Lastly, in section 3.3, we detail the form and information of the dataset created to tackle this problems in the following chapters. We also explain how we cleaned the dataset.

In Chapter 4 we will concentrate in the process of creating a data-driven hierarchy with the dataset presented in Chapter 3. Initially, section 4.1 introduces the concept of hierarchical time series. Then, section 4.2 explain the methodology applies to build the hierarchy and the results obtained wit each method.

Regarding Chapter 5 we will evaluate the performance of the different prediction models built in this chapter. First, section 5.1 explains coherent hierarchical forecasting models are made. In the following section, 5.2, we detail how we have constructed the predictor or predictors for each node in the hierarchy. Finally, section 5.3, is devoted to a thorough evaluation of all the prediction models that we tested and commenting the results obtained with each one.

Chapter 6 summarizes all the conclusions and lesson achieved from this project and outlines some future work directions.

CHAPTER 2
State of the art

In this chapter, we will first introduce the problem at hand, namely load forecasting, and then present different approaches to address it. Finally, we will come into contact with hierarchical forecasting models.

2.1 Load Forecasting

Forecasting is the process that involves making predictions based on past and present data and most often by analysis of the trends that exhibit the data itself. Forecasting is a problem that appears in a wide range of domains. For example, there are weather forecasting models to predict the humidity, temperature and wind velocity among other variables [17]; political forecasting to foresee instabilities on the government of different countries or outcomes of political events [9]; in economics to forecast situations of economical recession [20]; and in technology to prognosticate the new wireless communication technologies [3].

In the domain of electricity load forecasting, predictions are the power or energy that will be consumed, which are useful to find the balance of the demand and supply. In load forecasting the problem is about modelling a system who can accurately predict the consumption of a zone in the power grid. Depending on the scale, the consumption might come from just one household [22] to a country [4].

Load forecasting systems can be classified by the time-term or time horizon considered for the prediction, each one with different sets of objectives and applications. The most commonly used categories for this classification are the following:

- Short-term (from one hour to several days). It is used to aid power system operators on different decision making like planning, scheduling and so forth [14]. The most frequent objective of this type of decision making is the demand response for distribution and production of electricity [15].

- Mid-term (from one month to a season). It is mainly applied on the operational planning such as maintenance scheduling, planning for outages and major works in the power system [1] [2].

- Long-term (from one year or more). It is useful for grid expansion and its own operation [16]. Also, it has great importance from the business perspective and processes involving stakeholders, regulators and utilities [5].

2.2 Approaches

Electrical load forecasting is a problem that can be solved with a very broad range of techniques [15] like:

- Artificial Neural Networks (ANN), a machine-learning method based on the connections of neurons inside the human brain. This approach is well suited for load forecasting since it can provide good predictions when the data is incomplete or presents noise. Furthermore, it recognizes patterns and detects subtle relationships in the data. The main drawback of this approach is that it works similar to a black box, making it difficult to explain the relationships mentioned before. An example of this approach can be found in [10], where an ANN is applied to short-term forecasting in microgrid environments.

- Time series analysis (TSA), a statistical technique to manage a collection of data points organized in time. The main methods of this approach are the Autoregressive Integrated Moving Average (ARIMA) and Autoregressive Moving Average (ARMA). These models can work with stationary (not varying in time) and non-stationary data requiring just the past value of a time series. However, it needs a good comprehension of the intrinsic statistics and does not produce good long term predictions [27]. An instance of TSA that uses an ARIMAX model (the X comes from the use of exogenous information) for forecasting the power consumption of a building can be found in [19].

- Bottom-up systems, systems that are based on the construction of larger and complex systems by performing some kind of operation, often aggregation of elementary systems. The main advantage of this approach is that it considers the behaviour of different types of energy consumers. On the other hand, the lack of long term behaviour of the consumers makes it a model with poor results in the long run. An instance of this approach can be viewed at [8], where an stochastic bottom-up model for forecasting the load of an individual household is built.

- Support Vector Machines (SVM), a technique that constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space. The problem with this approach is that not all data can be perfectly split with a hyperplane and the computation in a high dimension space is very costly. The technique of soft-margins solves the first issue, while the Kernel trick method tackles the second one. Some authors tested this approach with two other approaches (Multiple linear regression and Multilayer Perceptron) to forecast at short-term in a non-residential building [18].

- Regression, a statistical method that measures the relationship between variables. There exists a wide variety of regression methods, what makes selecting the method that best fits the dataset the hardest aspect of this approach. An illustration of this approach can be seen in the work presented in [7], in which authors use a new regression model for hourly forecasting at long term.

2.3 Hierarchical forecasting

Besides the approaches seen in the previous section, a hierarchical variant has shown to be of interest to research, due to the fact that electrical load forecasts tend to exhibit some kind of hierarchical structure, as explained in the motivation. This commonly happens because the lowest levels of a hierarchy correspond to small sections of the grid, like buildings, and the highest levels to the total electrical consumption of a country [24, 23]. In other words, the upper levels typically represent the aggregation of the lower levels, even if the relationship between levels present different natures.

The relevance of using hierarchical information in the predictions can be observed in the fact that the Global Energy Forecasting Competition of 2017 focused its attention in hierarchical probabilistic load forecasting solutions, although models without hierarchical data were also provided [24]. Nevertheless, in the Global Energy Forecasting Competition of 2014, none of the participants used a hierarchical forecasting model [12]. This reveals the increasing interest on hierarchical models over the past years.

Works that rely on a hierarchical load forecasting typically use organizational hierarchies by geographical area because data are gathered from large geographical zones, and following a spatial criteria is more efficient. The reason behind this efficiency is that these forecasting systems tend to use exogenous variables, that is, information outside the system like weather data, which is useful to improve the overall accuracy. Hence, geographical organization turns out to be an adequate criteria to design a hierarchy when electrical consumption is highly dependent on factors such as the layout of the land and the weather conditions. In such a case, one can reason that consumers within the same geographical area have a similar consumption behaviour.

When the hierarchy can not be inferred with geographical information, different variants of clustering methods are applied as they are algorithms that group data based on their proximity. Another form of understanding this aggregation is that it organizes data grounded on the similarity, since samples that are closer to each other present similar values.

In addition to building the hierarchy, another step is necessary for the prediction model to work properly. When predictions are made at all nodes in the hierarchy, they must be coherent with each other; i.e., if a node represents the aggregation of its successor nodes then the prediction must be equal to the sum of the predictions of these nodes. There are two ways of doing this process:

- Best Linear Unbiased Mean Revised Forecasts, which is a method that produce coherent predictions by construction. This means that if we have the prediction at the top, the predictions at the lower levels are computed by disaggregating this one (top-down strategy), and that if we possess the prediction at the bottom level, then, the predictions of the aggregated nodes are in fact the sum of the lower level predictions (bottom-up strategy). Unfortunately, this has shown to give inaccurate results, especially when a large number of time series are involved [13].

- A two-phase approach, where predictions are first made at all levels of the hierarchy, and then, these are combined or reconciled to create coherent forecasts. The idea behind this approach is that it improves accuracy by synthesising information from different forecasts.

On the latter approach, the forecasting literature conveys two main lines of research:

- Mean hierarchical forecasting is characterized by predicting the mean value of the time series. On this line, it stands out the GTOP (Game-Theoretically Optimal Reconciliation) method [25]. The idea of GTOP lies in (1) calculating the best possible forecast for the time series disregarding aggregate consistency and then (2) use a reconciliation procedure to make the forecasts aggregate consistent. By explicitly separating the aggregate consistency from the forecast calculation, GTOP over-comes the limitations of the modeling power of bottom-up, top-down or middle-out methods, and allows considering estimators that use more complicated regression structures.

- Probabilistic hierarchical forecasting refers to a predictive probability distribution over future quantities or events of interest with the aim of improving the overall accuracy. In this regard, it sticks out the Mean Combined and Reconciled Probabilistic Forecasting of [23]. Using predictive distributions of the time series rather than mean forecasts only offers an important advantage, that of aggregating information from different levels in the hierarchy through a sparse forecast combination. Thus, using probabilistic forecasting of the time series enables quantifying the uncertainty in the predictions for the entire hierarchy while satisfying the aggregation constraints. In the end, this allows for better decision making and risk management.

CHAPTER 3

Problem statement and data processing

In this chapter we will first give a formal definition of the problem to solve. Then we will introduce the concept of day-ahead forecasting. Finally, the chapter presents the dataset used in this project.

3.1 Overview of the problem

The problem that we address in this project is the day-ahead hourly electrical load forecasting of the public structures managed by the council of Llíria. For this purpose, the local government of Llíria installed smart meters, devices that hourly record the electrical consumption on 69 public facilities.

Our proposal develops a solution to this problem which is divided into the following stages:

- Stage 1: Background of the problem. In this stage we investigate how the electricity market operates, specifically we delve into the special type of electrical load forecasting models known as *day-ahead forecast*. The explanation on how the day-ahead electricity market works, and the constraints and difficulties it poses is presented in section 3.2.
- Stage 2: Data processing. Our dataset is presented in section 3.3. One of the issues we need to deal with is the presence of missing values since the lack of values in the time series data give rise to information gaps of considerably significant size. Furthermore, the daylight saving time in which we turn clocks ahead or back, making us gain or lose one hour, causes non-normalized data and some adjustments are needed. This will be explained in detail in section 3.3.2.
- Stage 3: Prediction model. This stage is divided into two phases:
  - Hierarchical structure. The first phase involves building the hierarchical structure that follows a natural grouping of data, yielding a division of the load series in different levels. This structure will organize and present the dataset as a hierarchical time series, taking advantage of aggregated data. This phase is further explained in Chapter 4.
  - Design and Development of the prediction model. The second phase concerns the design and development of the forecast model that makes use of the hier-

archy built in the first phase. For this purpose, different models will be tested. The details of this phase are broken down in Chapter 5.

3.2 Day-ahead forecasting

The Day-Ahead Energy Market (day-ahead market) is a financial market where participants purchase and sell electric energy for the next 24 hours in a closed auction. Our problem thus consists in forecasting the electricity load of the next 24 hours using historical data. This is known in the literature as day-ahead forecasting and the reason behind its constraints is that the auction on which are going to be resourceful is held one day in advance of the actual values.

Day-ahead forecasting posses their own caveats and some added difficulties. The main difficulty is that the information that is needed or could help increase accuracy is not available at the time of the prediction, so the forecasting models tend to rely on other forecasts or lagged values to provide good predictions. For example, the climatological data that AEMET (Agencia Estatal de Meteorología or State Meteorological Agency) uses is not available until after two days have elapsed.

The auction of the electricity market is conducted to determine the price of the electricity one day before its consumption. In Spain, the market operator of the electricity market is OMIE (Operador del Mercado Ibérico de Electricidad or Iberian Electricity Market Operator). In this market, the two different types of participants, generators and consumers of electricity, submit their bids to take part in the trading. In our case, the council of Lliria is a consumer participant. The day-ahead market is performed every day, and the market participants have up to 12PM to submit their bids for each of the 24 hours of the following day.

3.3 Dataset

Llíria council runs several buildings and structures which are organized in 69 public facilities, a group of structures that produce a single electrical bill as they are considered as a unique entity. The nature of the public facilities is very different such as traffic lights, public offices, schools or markets. According to their type, we can reason that public facilities are organized around three major groups:

- Public buildings, facilities that are mainly used by public officials like counselors, office workers or school personnel. These edifices present a high electrical consumption in working hours.
- Street lightning. This type of structures show a low load consumption on daylight hours, since they are only activated at night.
- Special purpose constructions that present a unique consumption behaviour related to the activities undergone in the buildings. In this group we can find edifices like churches, penitentiary centers or Arab baths.

There is a smart meter installed in every public facility that hourly records the total electrical consumption of all the structures or group of buildings related to the facility. It is relevant to mention that as in any real scenario, some errors as downtime of the meters may occur, which provoke a lack of records for several hours of the load consumption. Also some facilities can completely cease its functions for extended periods of time, as for example public swimming pools that only opens in summer.

3.3.1. Dataset Creation

The smart meters deployed in each of the public facilities are identified by a key of 20 digits called CUPS (Código Universal de Punto de Suministro or Universal Supply Point Code). The meters record the hourly value of load consumption and the registered data is organized on a list for each month. In order to create a dataset with all the information, we designed a process that gathers all records in a single file and explicitly shows the date at which the event occurred. The date is inferred since we only have the list of values of each month with no information concerning the day and hour the values are recorded. This way, the first 24 read values correspond to the first 24 hours of the first day of the month. The rest of the values are computed with the same process, we only need to take care with the daylight saving, which implies that the day it occurs there is 23 recordings or 25 values, depending on the nature of the change.

The range of dates on which the load consumption is recorded is different for each smart meter. This is likely due to a gradual installation of the meters instead of setting them up all at once. In total, the data ranges from January 2016 to January 2019. Another explanation may be that some public facilities were not functional at the moment of the deployment, or that they stopped their activity before the record was finished since there also exist missing months at the end.

The dataset computed with the gathering presents the following features:

- the identifier of the smart meter (CUPS)
- the starting time of the 1-hour period
- the load in kWh for that period

Table 3.1 shows an excerpt of our dataset where each row represents the hourly consumption recorded by one smart meter.

CUPS	Date	Load
ES0021000008103001SY	23/02/2016 0:00:00	171170
ES0021000008103001SY	23/02/2016 1:00:00	170250
ES0021000008103001SY	23/02/2016 2:00:00	169920
ES0021000008103001SY	23/02/2016 3:00:00	170410
ES0021000008103001SY	23/02/2016 4:00:00	170630
ES0021000008103001SY	23/02/2016 5:00:00	170690
ES0021000008103001SY	23/02/2016 6:00:00	170110
ES0021000008103001SY	23/02/2016 7:00:00	135130
ES0021000008103001SY	23/02/2016 8:00:00	110
ES0021000008103001SY	23/02/2016 9:00:00	110
ES0021000008103001SY	23/02/2016 10:00:00	110
ES0021000008103001SY	23/02/2016 11:00:00	110
ES0021000008103001SY	23/02/2016 12:00:00	110
ES0021000008103001SY	23/02/2016 13:00:00	110
ES0021000008103001SY	23/02/2016 14:00:00	110
ES0021000008103001SY	23/02/2016 15:00:00	110
ES0021000008103001SY	23/02/2016 16:00:00	110
ES0021000008103001SY	23/02/2016 17:00:00	110
ES0021000008103001SY	23/02/2016 18:00:00	38040
ES0021000008103001SY	23/02/2016 19:00:00	190600
ES0021000008103001SY	23/02/2016 20:00:00	190800
ES0021000008103001SY	23/02/2016 21:00:00	192960
ES0021000008103001SY	23/02/2016 22:00:00	192690
ES0021000008103001SY	23/02/2016 23:00:00	176400

Table 3.1: Extract of hourly load consumption

Aside of the information collected by the smart meters, we also have at our disposal information about the public facilities where each smart meter was installed such as the location or the type of facility. This data will be helpful for us to better understand the consumption behaviour of each meter and validate the hierarchies produced in this project.

In this project we decided to use separately the data collected in winter and summer, thus, developing two forecasting models. This decision was made due to the fact that people consume electricity in different manners on each season of the year, being these two in particular the ones with more unique behaviours because of the temperature and hours of daylight. Winter has short cold days and summer has long warm days.

Although seasons are divided by four particular dates, corresponding with two equinoxes (date when the sun crosses the celestial equator) and two solstices (date when the sun reaches its maximum or minimum declination), we have established the division by months, taking into account the change of temperature in the province of Valencia. With these considerations in mind, the winter season takes up the months from December to February, and the season of summer the months from June to September. In table 3.2 we can see the two datasets created for the winter and summer season detailing the dates that were gathered along with the name given to each dataset.

Name	Period
WIN01	12/2016 - 02/2017 & 12/2017 - 02/2018 & 12/2018 - 01/2019
SUM02	06/2017 - 09/2017 & 06/2018 - 09/2018

Table 3.2: Dataset segmentation summary

We were obligated to work with only 37 out of the 69 public facilities due to a lack of recordings of load consumption over the winter and summer periods for some of the facilities. Nevertheless, the list of 37 facilities we worked with contain samples of the three group of public facilities mentioned above.

As we will see in the following chapters, with the available data, it was possible to perform interesting comparisons between the models as well as extract valuable conclusions at each step of the development of the project.

3.3.2. Normalizing and filling data

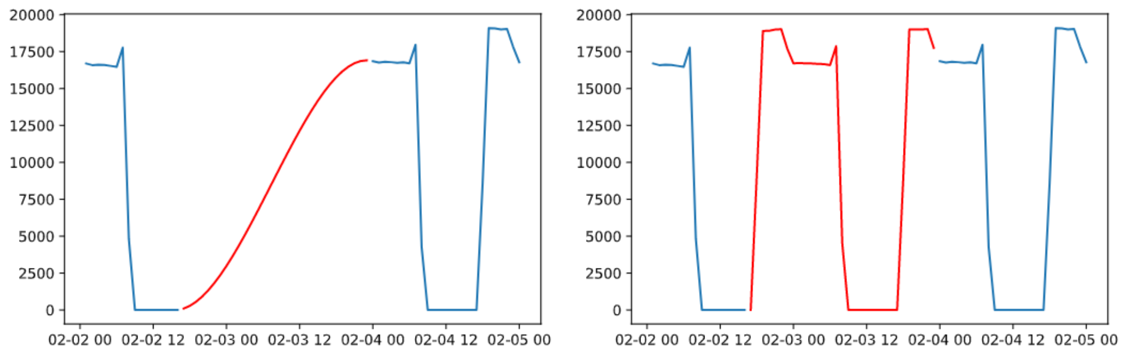
A problem that arises with the hourly load consumption is that, in Spain, we turn clocks one hour ahead or back from the current time in the last Sunday of March and October respectively. This causes non-consistent time series in which duplicates, in the sense that two different values have the same time of recording, or gaps in the information, appear. In order to solve this issue, we have decided to compute the average of the duplicated measurements that are produced in October, and fill out the gap of March as we will explain later in this section.

Some time series in the dataset also present gaps due to a meter malfunction, communication errors and power outages. In consequence, we will preprocess the dataset, filling out the gaps with different techniques. One of the most common techniques is to apply *interpolation* for missing values between a known period.

There exist different types of interpolation methods: linear, quadratic, cubic, spline, polynomial, etc. A simple test with each method provided us with the insight that spline and polynomial interpolation provide better approximations to the "correct" value since they tend to calculate a smooth curve that resembles more closely the actual behaviour of data.

However, using interpolation directly on the time series does not solve the problem as it only fills in the gaps with a curve line that does not maintain the seasonality of the data. With the aim of improving the interpolation, we deseasonalized the time series by interpolating only the load consumption of the same hour, in other words, we split a time series into 24, being each one a series composed by the values of one of the 24 hours. Then, we can interpolate the missing values on each of the 24 time series. Afterwards, we add the interpolated values to the original time series to fill the gap. This approach is a better approximation since the time series formed by the values on the same hour have a more uniform behaviour and the gaps of the deseasonalized time series are smaller, thus providing interpolated values that respect the seasonality of the time series.

Figure 3.1 shows both interpolations explained before; the one of the left (Figure 3.1a) is the approach wherein the seasonality is not preserved; the figure on the right (Figure 3.1b) is the approach in which we deseasonalize the time series to compute the missing values. The X-axis represents the time that ranges From 2nd February to 5th February, and the Y-axis is the consumption in KWh. The blue plot denotes the existing recorded values the red plot is the reconstructed values computed with interpolation. It is important to mention that the consumption values are indeed points, since they are hourly values, but they have been linked by a line to properly show the seasonality of the data so that the blank spaces are not missing values.



(a) Without deseasonalize **(b)** With deseasonalize

Figure 3.1: Dataset filling with interpolation

Finally, the second approach in which we deseasonalize the time series presents an edge case which leaves small gaps. These small gaps are in fact the endings of the same-hour time series, as when they are seen in the deseasonalized time series it is not between two known points, however, in the original time series the gap may be between two known values. To fill these gaps, we used the first approach, we interpolated the time series without deseasonalizing, since they are only one-value gaps with a small margin of error.

CHAPTER 4
Hierarchy

This chapter will focus its attention on the process and results of building a data-driven hierarchy. First, we will introduce the concept a hierarchical time series. Secondly, we will explain the used methodology, and then we will analyse the results provided by three different ways. Finally, we will extract some conclusions of the results.

4.1 Hierarchical time series

In many occasions, the time series of a particular subject can be grouped following some criterion, creating another time series by aggregating the time series that form the group. This process ends up forming a time series hierarchy, also known as a hierarchical time series (HTS). At the same time, a HTS can be formed by disaggregating a time series. These two types of processes to produce an HTS are known as:

- Bottom-up: when the hierarchical time series is built by aggregating the time series at the lowest level (base time series) till arriving to a time series that is the aggregation of all the ones below it.
- Top-down: when we start from the time series at the highest level of the hierarchy and we split it to form the lower levels till reaching the base of the hierarchy.

These hierarchical time series often arise due to geographic divisions. For example, the Valencian Community can be disaggregated by provinces, and these provinces can be divided in localities as can be seen in Figure 4.1.

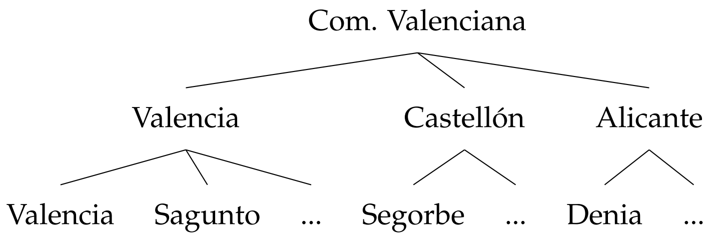


Figure 4.1: A hierarchical load time series by region

In our case, we depart from the time series produced by each smart meter, therefore, a bottom up approach has been performed in which the grouping is produced by the similarities of the data as it will further explained in the next section.

4.2 Clustering

A data-driven hierarchy is built by automatically finding commonalities in the data of different elements (load series in our case) and grouping them together. Hence, the nature of the inferred hierarchy is ultimately dependant on the information used to represent each element. The first step in our data-driven approach is to define a feature vector, a collection of features that represent an element in our hierarchy.

For our part, we reason that the difference between the load series lies on the usual hourly consumption of a day, so our feature vector would be formed by the average hourly consumption of all days, that is to say, the feature vector is composed by the average consumption of each hour (0, 1, ..., 23) across all the days. This feature vector can then be understood as a "consumption profile", that shows how the consumption varies during the day.

This consumption profile provides a characterization of the time series that allows to group them in a similar manner to that expected in the previous chapter, in which three groups of public facilities where identified without a deep analysis. Nevertheless, other two feature vectors were tested. In total, the features vector that were used are the following:

- Day Profile feature vector: daily average of hourly consumption.
- Week Profile feature vector: weekly average of daily consumption, a feature vector composed by seven features, each one being a day of the week (Monday, Tuesday, Wednesday, etc.).
- Concatenated feature vector: the union of the two previous profiles, therefore, it presents 31 features, 24 of the day profile and 7 of the week profile.

It is also relevant to note, that each feature vector is normalized, that is to say, the values are divided by the maximum value of the vector. It is important to normalize the data since we want that all dimensions or features are treated equally.

With regard to the data-driven hierarchy, we propose the following 3-tier structure as a first approach:

- Tier 1: this is the lower tier and each element in this tier represents the electricity load of a public facility of Llíria as recorded by a smart meter. Therefore, we will have as many tier-1 elements as smart meters in our dataset.
- Tier 2: the mid tier in the hierarchy represent groupings of load series that display a similar consumption profile. The number of elements in this tier and groupings will be automatically inferred from the data, hence the data-driven nature of this hierarchy.
- Tier 3: the higher tier and single root of the hierarchy represents the total consumption of the public facilities under the Lliria Townhall.

The unsupervised method to group a collection of data is known as clustering. There are many algorithms that perform clustering but we are going to test the feature vectors with the following ones: K-Means clustering, Mean-Shift clustering and Expectation-Maximization (EM) clustering using Gaussian Mixture Models (GMM) [21].

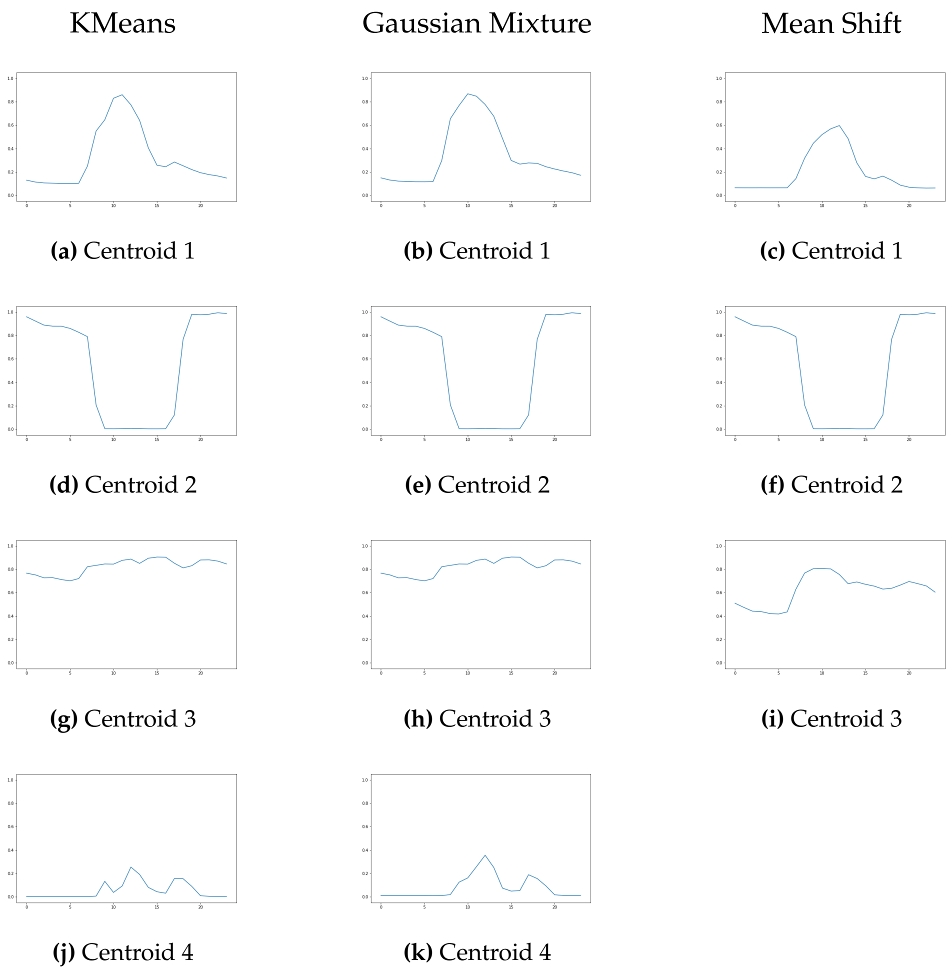
K-Means clustering is an algorithm that given a number of clusters or classes tries to minimize the sum-of-squares (the difference between a point and the average of the class squared) inside the same cluster. Mean shift clustering is a sliding-window-based algorithm that attempts to find dense areas of data points. It works by updating centroids to be the mean of the points within the sliding-window. In GMM it is assumed that data points are Gaussian distributed, thus, the shape of the clusters are defined by two parameters, the mean and the standard deviation. In order to find this parameters, the expectation-maximization algorithm is applied.

K-Means and GMM algorithms have as a parameter the number of clusters that it has to make, thus, to calculate the goodness of a configuration of clusters we decided to compute the 'rule of elbow', a method that consists of plotting the sum of squares distance to each centroid (average sample of the cluster) as a function of the number of clusters and picking the 'elbow' of the curve as the number of clusters to use.

4.2.1. Day profile feature vector

The day profile feature vector represents the average consumption behaviour on a day, thus, with this feature vector we expect to find groups that consume on a daily basis in a similar way.

In the dataset WIN01, KMeans and GMM return an optimal configuration of four clusters while Mean Shift produces only three, as seen in Figure 4.2. The four clusters produced by the two first algorithms are very similar, varying on only a few samples.



(a) Centroid 1 **(b)** Centroid 1 **(c)** Centroid 1

(d) Centroid 2 **(e)** Centroid 2 **(f)** Centroid 2

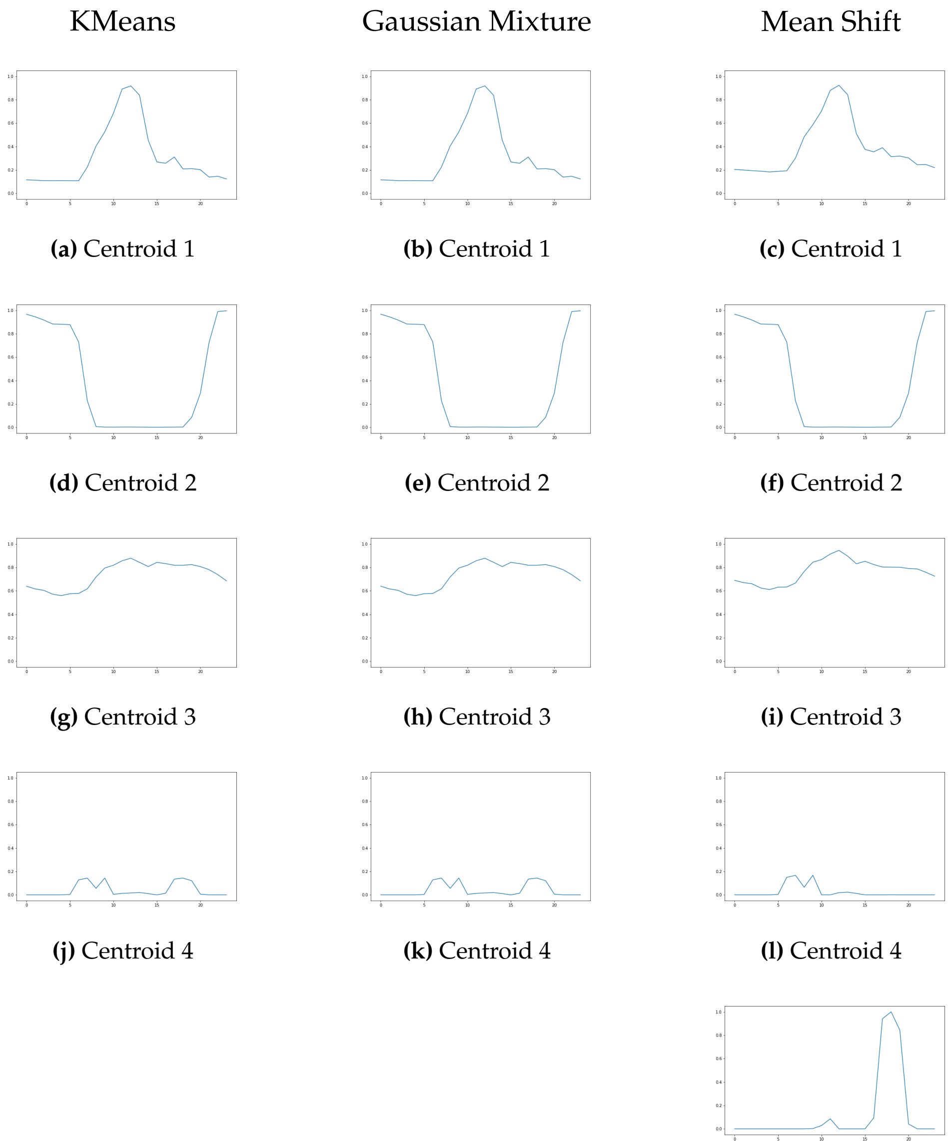
(g) Centroid 3 **(h)** Centroid 3 **(i)** Centroid 3

(j) Centroid 4 **(k)** Centroid 4

Figure 4.2: Winter Day Profile clustering and centroids

These clusters have the following characteristics:

- First cluster (Centroid 1): public facilities that have always some consumption but present a higher load consumption on daylight hours or work hours. In this cluster we find mainly public buildings.
- Second cluster (Centroid 2): public facilities that only consume on night hours. It consists of almost all street lightning facilities.
- Third cluster (Centroid 3): public facilities that show a nearly constant consumption. This cluster has traffic lights and pumping stations.
- Fourth cluster (Centroid 4): public facilities that have some low consumption at midday. This is the cluster with the most different kind of buildings, thus, we reason that they are grouped because they have very little activity. We can find an old hydraulic cereal mill, a nursing home and a food bank among other buildings.



(a) Centroid 1 **(b)** Centroid 1 **(c)** Centroid 1

(d) Centroid 2 **(e)** Centroid 2 **(f)** Centroid 2

(g) Centroid 3 **(h)** Centroid 3 **(i)** Centroid 3

(j) Centroid 4 **(k)** Centroid 4 **(l)** Centroid 4

(m) Centroid 5

Figure 4.3: Summer Day Profile clustering and centroids

When it comes to the results of Mean Shift algorithm, it looks like the samples of the fourth cluster that produce the other two algorithms have been integrated in the first centroid (c) and third one (i), producing a variation on the centroid.

We found similar results in the dataset SUM02 as shown in Figure 4.3. This time, however, Mean Shift returns five clusters instead of three. The centroids appear to be affected by the change of daylight hours, since for example, the cluster of street lightning (Centroid 2) has increased the hours of no consumption or the cluster of public buildings (Centroid 1) has a narrower peak. The fourth cluster (Centroid 4) shows that the activities consuming electricity are now done before or after midday, probably because these are the hottest hours of the day. In the case of Mean Shift it seems that the fourth cluster (l) has been splitted to produce the fifth one (m), since the samples that produced the consumption after midday have been moved to the latest. This cluster (m) show a peak because in average, the samples have its highest consumption in these hours, but when they were unified with the cluster (l) showing a centroid like the fourth one of KMeans (j), the peak was attenuated by the other samples, since they have almost no consumption.

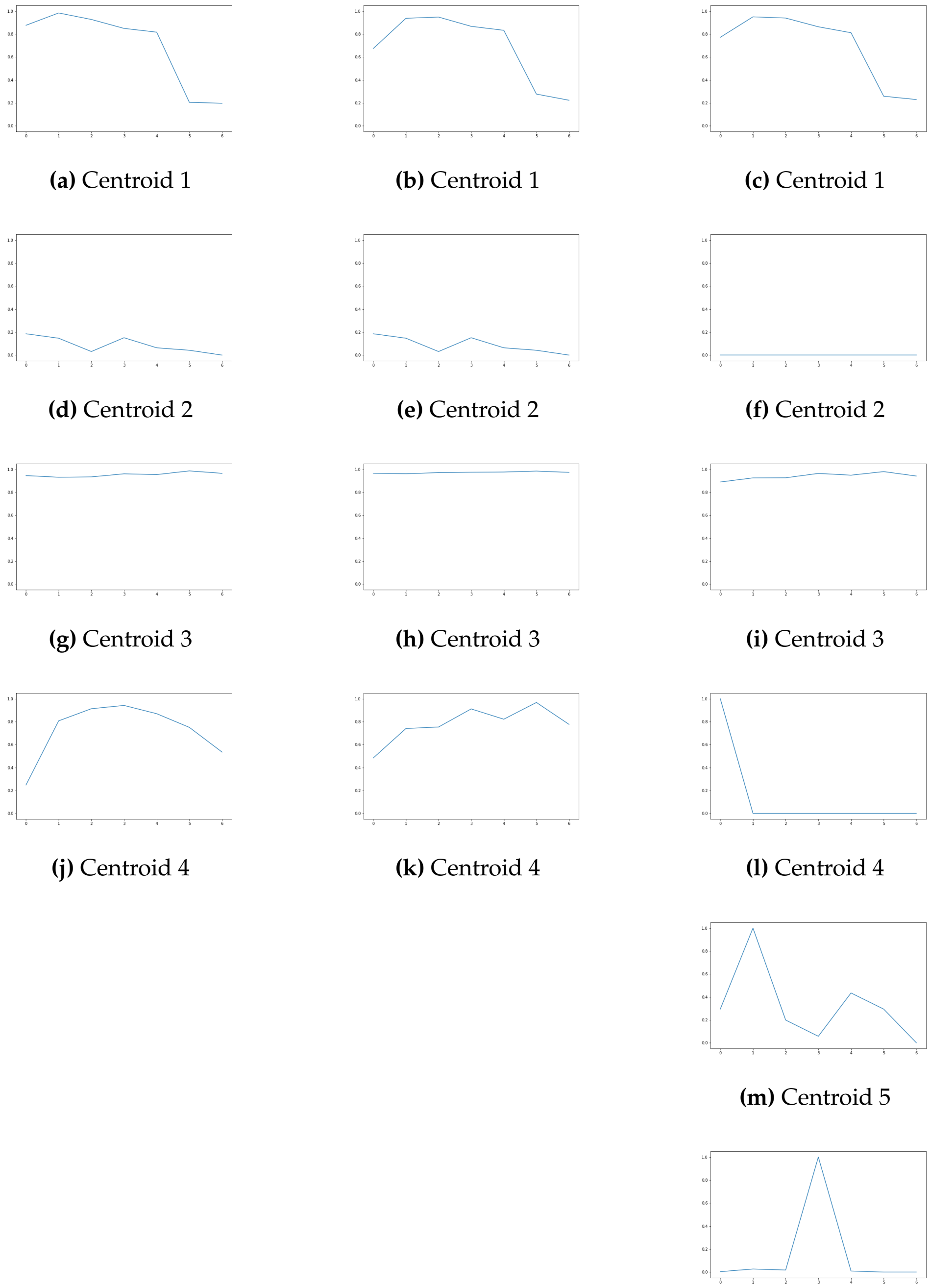
4.2.2. Week Profile feature vector

The week profile feature vector represents how we consume electricity in each day of a week, providing a behaviour that is affected in particular by the weekends and showing in which days there is more activity.

On the first hand, in the dataset WIN01, the algorithms of KMeans and GMM have an optimal configuration of four clusters, contrarily, Mean Shift returns six groups. The centroids of these clusters can be seen in Figure 4.4. As with the day profile, the four clusters created by the two first algorithms are quite similar. They present the following behaviour:

- First cluster (Centroid 1): public facilities that show a high consumption during working days and a drop on the weekends. It is formed by public offices.
- Second cluster (Centroid 2): public facilities that have some consumption specially on Mondays, Tuesdays and Wednesdays. It is composed by public facilities of very different types.
- Third cluster (Centroid 3): public facilities that have an almost constant consumption. This time, we can find street lightnings merged with the traffic lights and the pumping station, since seen from the perspective of a week, they consume persistently.
- Fourth cluster (Centroid 4): this cluster differs a little between KMeans and GMM, but in general it represents public facilities that have a growing consumption at the beginning of the week and then decreases over weekends. In the case of KMeans this cluster is formed by some public buildings like a mausoleum and a museum. When it comes to GMM this cluster is only consists of a sample, a market.

It is relevant to mention that the last three clusters produced by Mean Shift (l, m, n) are composed by only one public facility, thus, it gives us the insight that this algorithm does not perform well with this profile. The three public facilities are: a housing, a market and a food bank.



KMeans | Gaussian Mixture | Mean Shift

(a) Centroid 1 **(b)** Centroid 1 **(c)** Centroid 1

(d) Centroid 2 **(e)** Centroid 2 **(f)** Centroid 2

(g) Centroid 3 **(h)** Centroid 3 **(i)** Centroid 3

(j) Centroid 4 **(k)** Centroid 4 **(l)** Centroid 4

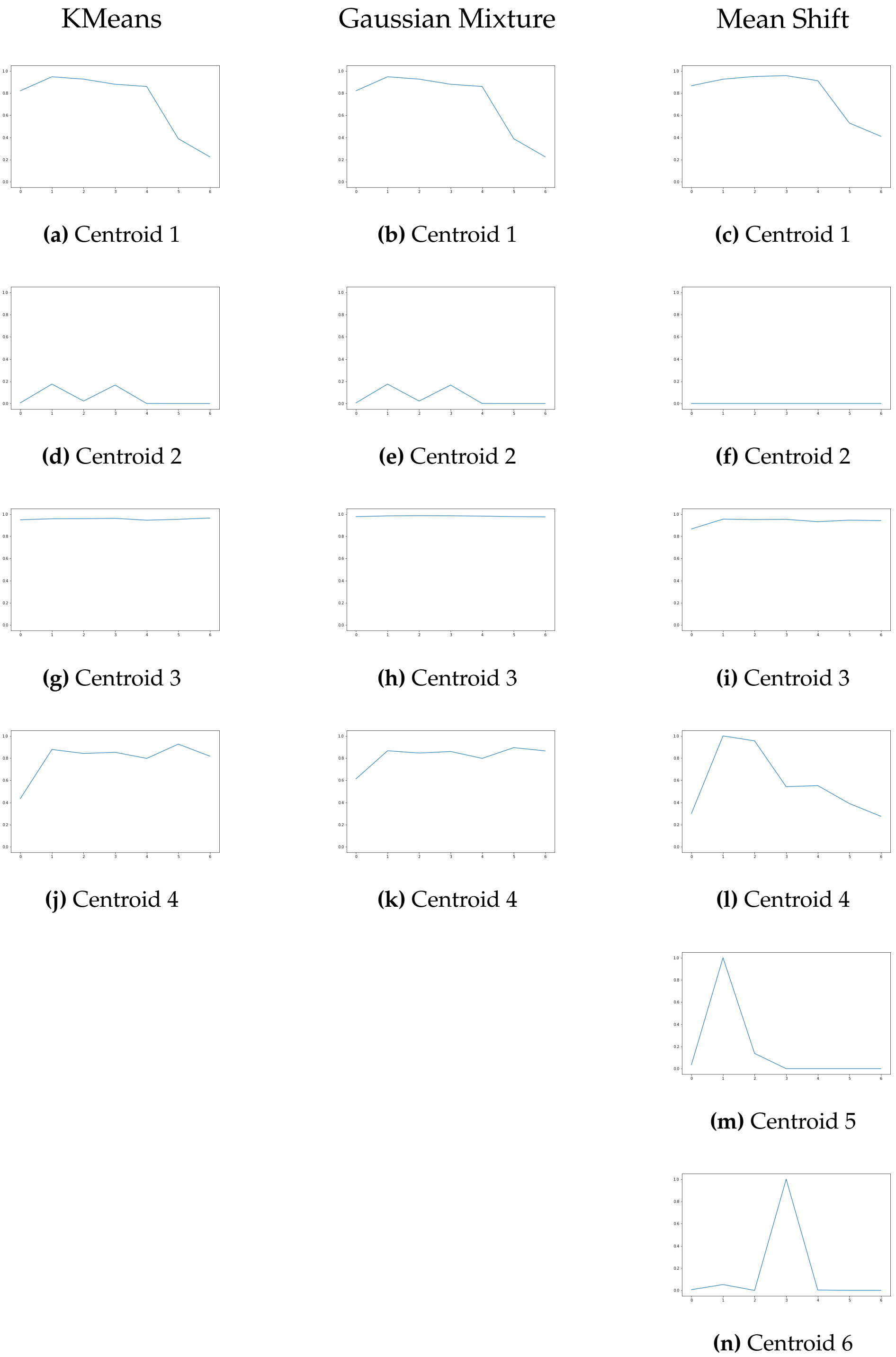
(m) Centroid 5

(n) Centroid 6

Figure 4.4: Winter Week Profile clustering and centroids

On the other hand, in summer, the first two algorithms have also established an optimal configuration of four clusters, as can be seen in Figure 4.5. As with the previous profile, the change on temperature and daylight hours has changed the consumption behaviour, although this time is more subtle, since it affects all the days equally. The fourth cluster (Centroid 4) is the one that shows more variation, loosing the decreasing consumption of the weekends.

In reference to Main Shift, as with the winter season, the last three clusters (l, m, n) are formed by one sample, although, one of them has changed, the housing has been switched by a public office.



(a) Centroid 1 (b) Centroid 1 (c) Centroid 1
(d) Centroid 2 (e) Centroid 2 (f) Centroid 2
(g) Centroid 3 (h) Centroid 3 (i) Centroid 3
(j) Centroid 4 (k) Centroid 4 (l) Centroid 4
(m) Centroid 5
(n) Centroid 6

Figure 4.5: Summer Week Profile clustering and centroids

4.2.3. Concatenated feature vector

The concatenated feature vector is the union of the day and week profile and hence we can divide the centroids into two parts, the one that represents the day profile and the one that refers to the week profile. Regarding the results of clustering, there are certain aspects that are important to mention. The results obtained with the dataset WIN01 can be seen in Figures A.1 and A.2, and the outcomes with the dataset SUM02 are shown in Figures A.3 and A.4, which form part of the Annex A.

In the first place, all three algorithms have an optimal configuration of four clusters, giving us the insight that with this feature vector there are clear separations between four groups. These groups are almost identical to the ones formed with KMeans in the day profile feature vector.

- First cluster (Centroid 1): public facilities that have always some consumption but present a higher load consumption on daylight hours or work hours.
- Second cluster (Centroid 2): public facilities that only consume on night hours.
- Third cluster (Centroid 3): public facilities that show a nearly constant consumption.
- Fourth cluster (Centroid 4): public facilities that have some low consumption at midday.

Secondly, since the four groups are similar to the ones of the day profile feature vector, we reason that this is due to the features referring to the day profile have more weight than the ones related to the week profile, in part, because we are facing 24 features to 7. This can be seen in the centroids of the week part, as two clusters with almost constant consumption can be found, one formed by the street lights and the other by the traffic lights, pumping stations, etc. To solve this problem some kind of projection to an equal number of features should be performed.

4.2.4. Conclusions

After analysing all the results given by the three clustering algorithms, we decided to use KMeans as the method to build the hierarchy of our forecasting models. This decision is supported by the fact that the three of them produce similar clusters, but only KMeans is deterministic, that is, before a same dataset, it produces always the same clusters, thus, providing us more confidence than the other two algorithms. Furthermore, Mean Shift and Gaussian Mixture tend to produce clusters with only one sample.

With that said, our data-driven hierarchy will present four nodes or groups at the middle level, although, depending on the type of feature vector, the children or samples forming that groups will be different. An example of this hierarchy can be found in Figure 4.6.

On the other hand, at this point we cannot make a decision on the profile that will perform better for the forecasting model. Consequently, the hierarchies produced by the three different feature vectors will be tested on the following chapter.

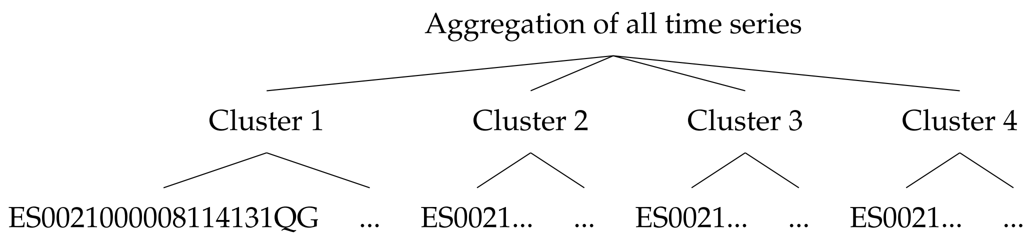


Figure 4.6: A data-driven hierarchy obtained by clustering

CHAPTER 5
Prediction model

This chapter is devoted to evaluate the different prediction models built. First we will explain the characteristics of the coherent hierarchical forecasting and how it can be done. In second place, we will formalize and explain the models that we have created. Then, we will discuss the results obtained. At the end, we will present some conclusions extracted from the results.

5.1 Coherent Hierarchical Forecasting

In hierarchical forecasting, the predictions made at each level of the hierarchy, also called base predictions, are usually not coherent between them because the predictions of a time series are independent of another one. Coherence on hierarchical forecasting means that the relation established between levels is fulfilled. That is, that the predictions of a node are equal to the aggregation of the node children's predictions. As we mentioned in section 2.3, there are two ways of producing coherent hierarchical forecasts:

- Best Linear Unbiased Mean Revised Forecasts, a method in which the predictions are constructed following a top-down or bottom-up strategy.
- A two-phase approach in which a prediction for each node is made and then predictions are reconciled.

For our part we decided to test only the first approach using two different strategies: (1) a bottom-up strategy, which simply builds the aggregation predictions following the hierarchy, and (2) another strategy that follows the minT methodology [26], which involves a few more steps and produces a more sophisticated model.

In Best Linear Unbiased Mean Revised Forecasts, the predictions that comply with the aggregation constraints of the hierarchy take the following form:

$$\tilde{y}_t = SP\hat{y}_t$$

where $\hat{y}_t$ are the base predictions made independently of the aggregation constraints, and $\tilde{y}_t$ are the predictions reconciled, thus fulfilling with the constraints. Aside of these terms, S and P which are both matrices. The matrix S is known as the summing matrix since it represents the aggregation constraints and establishes how the predictions at the bottom level must be added to form the other predictions at higher levels of the hierarchy. The matrix P is recognized as the coherency matrix as it maps the base predictions to the reconciled predictions of the bottom level, which means that the reconciled predictions take in consideration the predictions made at the intermediate levels of the hierarchy.

All things considered, the difference between the simple bottom-up strategy and the minT lies in the definition of the matrix P as for the same hierarchy the summing matrix S remains equal.

In order to explain how both strategies have been implemented we will start with an example of a two-tier hierarchy which can be seen in Figure 5.1. At the higher level we can find the total aggregation of all time series, which we will call *Total*. The values or observations of a time series are denoted by the symbol y and so the observation at time t at the *Total* time series is $y_{Total,t}$. The top tier time series is disaggregated into two time series, the A series and B series. Finally, both time series are disaggregated into three and two time series respectively. Then, the total number of time series is $n = 8$ and the number of series at the lowest level (leaf nodes) is $m = 5$.

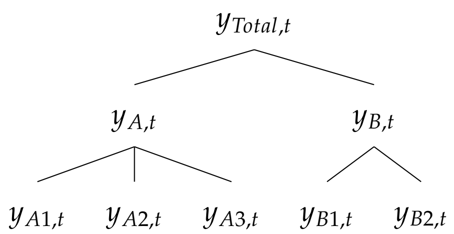


Figure 5.1: Example of a two-tier hierarchy with five leaf nodes

Once the hierarchy has been established, we first build a summing matrix of order $n \times m$ (n = total number of time series and m = number of time series at the bottom level). It is important to mention that the m columns represent the nodes at the bottom level of the hierarchy ($y_{A1,t}$, $y_{A2,t}$, $y_{A3,t}$, $y_{B1,t}$, $y_{B2,t}$), and the n rows represent the nodes in the hierarchy ordered by its level: first the top node, then the first child of the top node, then the second child and so on up to the last nodes that belong to the bottom level ($y_{Total,t}$,$y_{A,t}$,$y_{B,t}$$y_{A1,t}$, $y_{A2,t}$, $y_{A3,t}$, $y_{B1,t}$, $y_{B2,t}$). The matrix S is constructed following these steps:

1. First we compute n - m rows that represent the aggregation constraints. The n - m nodes that are not at the lowest level of the hierarchy are the nodes formed with the aggregations of the nodes below them. Each cell of the row is filled with a 1 if the node corresponding to that column is below the node corresponding to the row itself, and 0 if not. For example, the row corresponding to $y_{A,t}$ is [1 1 1 0 0].

2. On second place, we fill the rest of the matrix S with an identity matrix of order m, since each node at the bottom is, in fact, an aggregation of itself with zero.

Following the hierarchy of the example in Figure 5.1, the result summing matrix would be:

$$S = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ & & I_5 & & \end{bmatrix}$$

5.1.1. Bottom-up

In the simple bottom-up strategy, the coherency matrix (P) is constructed quite simply as $P = [0_{m\times(n-m)}|I_m]$, where $0_{m\times(n-m)}$ is a null matrix (filled with zeros) and I_m is the identity matrix. When the matrix takes this form, the reconciled predictions are in fact the same as the base predictions of the bottom level, since the values in other levels of the hierarchy are not considered. In consequence, the predictions in the hierarchy are, as we explained, just the aggregation of the base predictions of the bottom level nodes, and thus, they are coherent. In the example of the Figure 5.1, the coherence matrix would be:

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

5.1.2. MinT

In the case of minT, Wickramasuriya et al. [26] explained different methods to estimate the matrix P given that it depends on the forecasts already reconciled and so it can not be computed. Actually, the coherency matrix P that would lead us to the reconciled forecasts is given by the following formula:

$$P = (S'\Sigma_h^{-1}S)^{-1}S'\Sigma_h^{-1}$$

where Σ_h is the covariance matrix of the h-step ahead reconciled forecasts errors. This matrix is not identifiable as Wickramasuriya et al. [26] revealed, although we can compute the optimal matrix P by:

$$P = (S'W_h^{-1}S)^{-1}S'W_h^{-1}$$

where W_h is the covariance matrix of the h-step ahead base prediction errors, and it is subject to the condition that $SPS = S$. Wickramasuriya et al. [26], presented four different simplifying approximations to calculate this matrix. For our part, we have decided to use the following methods:

- OLS (Ordinary Least Squares). This approximation considers that the matrix P is independent of the data and that it only depends on the summing matrix S.
- MinT-VarScale or WLSV (Variance Weighted Least Squares). This approximation scales the base forecasts at all levels, by employing the variance of the difference between the observed value and the mean value of the predictions for that observation, also called residuals.

We used the implementation of both methods provided by the library *scikit-hts* in Python.

5.1.3. Base predictions

Despite the wide range of techniques for building a base predictor presented in section 2.2, in this project we decided to use time series analysis for producing the predictors of each node in our hierarchy. As explained before, the two main methods for time series forecasting are the ARIMA and ARMA models. For our part, we used a variant of the ARIMA model, the SARIMA (Seasonal AutoRegressive Integrated Moving Averages) model, more particularly the implementation of the library *statsmodels* in Python. The difference between ARIMA and SARIMA lies in that the latter can tackle seasonality, as its name suggests.

SARIMA models require two components formed by a total of seven parameters:

- Order (p, d, q):
  - p: auto-regressive (AR) models forecast the next value or point in a time series by taking into account the previous values and using them to compute a mathematical formula that is closely related to a linear regression. The parameter p is an integer number that describes how many previous values will be used for the prediction.
  - d: the integration part (I) of the SARIMA models represents that data must be stationary, that is to say, it does not have to present seasonality or trends. To do so, the difference of the time series is taken by subtracting the previous values from each value in the series. This computation has a propensity to produce a stationary series, or at least more stationary that it was before. The d parameter is an integer numeral that denotes how many times the series has to be differentiated.
  - q: a moving-average (MA) model executes some computations based on noisy data along with the slope to carry out a short autocorrelation. The parameter q is an integer value that stands for a similar meaning than the auto-regressive one, being how many previous values of the series are putted into the equation.
- Seasonal order (P, D, Q, s):
  - (P, D, Q): these parameters are equivalent to the ones in Order (p, d, q), the only difference being that P, D and Q only apply to the seasonal part.
  - s: the s parameter is an integer number that indicates the periodicity of the seasonal cycle of the data. For example, if the values are split on a monthly basis, the seasonal cycle is a year and the value of s is 12. If the data points are separated on a daily basis, the seasonal cycle is a week and the value of $s = 7$.

5.2 Forecasting models

With the purpose of testing the bottom-up strategy and the one that follows the minT methodology, we have built three forecasting models for two different variants; therefore, we have constructed a total of six forecasting models. Each variant has its own characteristics, since the prediction step or the interval of values that have to be predicted ahead of the current time differ from one variant to another.

5.2.1. First variant: one predictor per node

The first variant establishes three models that have one predictor per each time series used, but for simplifying we will say that there is one predictor per node.

It is important to notice that with this variant, the prediction step ranges from 24 to 48, in other words, the interval predictions that we are interested go from 24 to 48 from the actual time if we are dealing with a day-ahead forecasting. This is better understood with an example: if we want to predict the 24 values of tomorrow, as it is done in day-ahead forecasting, we have to start from the last value of yesterday since the present day has yet not begun, and thus, we do not have its values. With that said, we have to skip the first 24 predictions, which represent the current day, and keep the next 24, which are related to the day of tomorrow. In consequence, we are taking the values at order 24 to 48 of the predictions.

These three models present the following features:

- BU_1: a model that follows a bottom-up approach of the Best Linear Unbiased Mean Revised Forecasts methodology; it only needs one predictor for each of the bottom level nodes, i.e., 37 predictors in our case.
- $root_1$: this model presents only the predictor of the root or top node; that is, the node that represents the aggregation of all time series.
- $minT_1$: in this forecasting model we have built one predictor for each node in the hierarchy so we have 42 predictors (37 nodes at the bottom, 4 at the intermediate level and 1 at the top). Furthermore, we reconcile the predictions by performing both OLS and WLSV.

The base predictions of this variant employ a SARIMA model with the default Order $(1,0,0)$ and a Seasonal order of $(1,1,1,24)$, since our data is arranged in an hourly basis and in consequence the seasonal cycle is a day (24 hours).

5.2.2. Second variant: 24 predictors per node

The second variant differs from the first one in that each node has 24 predictors, a predictor for each hour of a time slot of a day. In fact, it is the same process that we performed in section 3.3.2 for deseasonalizing the time series.

Another relevant thing to mention is that, as we divided the time series by the time slot, the prediction step is smaller. In other words, with the first variant we had to predict with the same predictor the next 24 to 48 values while in the second variant we only have a prediction step of two, since we skip the first next value, that would represent the prediction of the current time at a specific time slot, and the second prediction refers to the prediction of tomorrow.

The three models of this variant are arranged in a similar manner than the first variant, being:

- BU_{24}: the model that follows the bottom-up approach of the Best Linear Unbiased Mean Revised Forecasts methodology.
- $root_{24}$: the model in which we only predict for the root node.
- $minT_{24}$: the model that reconciles the predictions of all nodes in the hierarchy.

With this variant, the SARIMA models of the base predictions present a seasonality of s = 7, since each values would represent a day, and in consequence, the seasonal cycle would be a week. We have used also the default Order $(1,0,0)$, and the Seasonal order would be as follows $(1,1,1,7)$.

5.2.3. Summary

In Table 5.1 we can see a summary of all the models explained with the total number of predictors used, which will help to show the magnitude of each model.

Model	Predictor x Node	Total Nº Predictors	Description
BU_1	1	37	Bottom-up
$root_1$	1	1	Root prediction
$minT_1$	1	42	Reconciled model
BU_{24}	24	888	Bottom-up
$root_{24}$	24	24	Root prediction
$minT_{24}$	24	1008	Reconciled model

Table 5.1: Summary of forecasting models

5.3 Experimental Evaluation

To test the six models that we have built for this project, we decided to use as dataset only the season of winter and summer of one year. The season of winter comprises from December of 2017 to January of 2018 and the season of summer ranges from June of 2018 till September of 2018. This year was taken as it is the most recent and complete one. We also had for the winter season some data of the year of 2019, but it lacks the month of February, thus, the season would comprise only two months instead of three.

Both datasets have been divided in two sets of data, the training and validation set. The training set has been used to build the predictors of the model, and the validation set has served us to measure the accuracy of each model. The split between both set has been made so that the validation takes the last 10 days of the season. In Table 5.2 we can see a summary of both sets with the number of values that have each one and its identification name.

Season	Name	Set	Date	Nº Values
Winter	WINPRED01	Training	01/12/2017 - 18/02/2018	1920
		Validation	19/02/2018 - 28/02/2018	240
Summer	SUMPRED02	Training	01/06/2018 - 20/09/2018	2688
		Validation	21/09/2018 - 30/09/2018	240

Table 5.2: Summary of training and validation sets

For evaluating the accuracy of the models, we decided to use three well known metrics for forecasting problems on the predictions of the root or top node, since our interest is to accurate predict the total consumption. The metrics are:

- ME (Mean Error): it is the mean of the differences between the values observed and the value predicted. It is also known as the forecast bias, since it indicates if the model tends to under-forecast (negative value) or over-forecast (positive value). The formula is:

$$ME = (\frac{1}{n}) \sum_{t=1}^{n} y_t - \tilde{y}_t$$

- MAE (Mean Absolute Error): it is the same as ME with the difference that the error is in absolute value, so that negative values do not cancel the positive ones and vice-versa. It shows how much fails to predict the model. The formula is:

$$MAE = (\frac{1}{n})\sum_{t=1}^{n}|y_t - \tilde{y}_t|$$

- MAPE (Mean Absolute Percentage Error): it is quite similar to MAE, the main difference is that it is scale-dependent since we divide the MAE with the actual value, thus, giving a percentage. There exist some caveats that are relevant to mention. First, the intuition says that the values range from 0% to 100%, but it can be greater. In second place, it cannot be used when the actual values are 0 as we cannot divide by this value. This is one of the reason that it is difficult to use in load forecasting, as there may be no consumption. In our case it is not a problem because we only evaluate with the observations of the top node, that in any time the value is 0. The formula is:

$$MAPE = (\frac{1}{n})\sum_{t=1}^{n}\left|\frac{y_t - \tilde{y}_t}{y_t}\right|$$

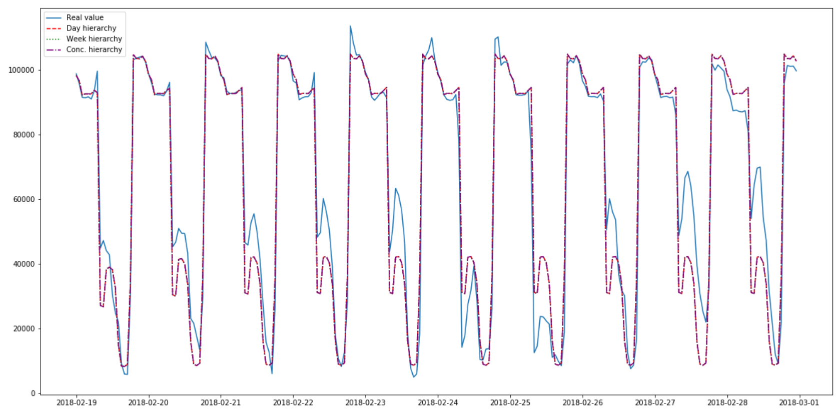
5.3.1. Winter season results

The results of the models with the dataset of WINPRED01 can be seen in Table 5.3. At first glance, the models that have one predictor per node (BU_1, $root_1$, $minT_1$) have less accuracy, considering the MAPE error, than the models that present 24 predictors (BU_{24}, $root_{24}$, $minT_{24}$), manifesting almost 10% of error less. This gives us the insight that when the prediction step is smaller, i.e., predicting closer from the known values, there is less room for error and the accuracy of the base predictions increases. This improvement in the accuracy of the base predictions has an impact in the model, improving also the accuracy of the model.

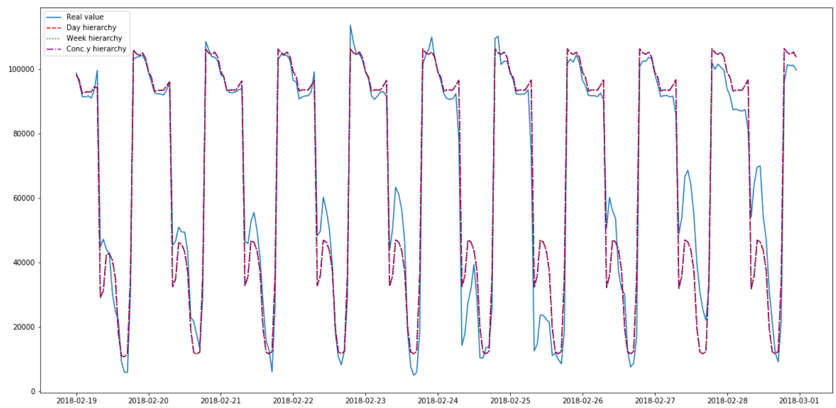
Model	Reconciliation	Hierarchy	ME	MAE	MAPE
BU_1	None	Not relevant	772.8061	6177.7572	0.2033
$root_1$	None	Not used	-1977.6650	6418.1765	0.1873
$minT_1$	OLS	Day	-1905.2842	6402.9428	0.1872
		Week	-1905.2842	6402.9428	0.1872
		Conc.	-1905.2842	6402.9428	0.1872
	WLSV	Day	60.9806	6166.1484	0.1952
		Week	60.9806	6166.1484	0.1952
		Conc.	60.9806	6166.1484	0.1952
BU_{24}	None	Not relevant	1649.7941	4180.2191	0.1233
$root_{24}$	None	Not used	1893.8011	4076.1108	0.1193
$minT_{24}$	OLS	Day	1887.3799	4051.1479	0.1183
		Week	1887.3799	4051.1479	0.1183
		Conc.	1887.3799	4051.1479	0.1183
	WLSV	Day	1734.7404	3885.4117	0.1111
		Week	1734.7404	3885.4117	0.1111
		Conc.	1734.7404	3885.4117	0.1111

Table 5.3: Winter results

Regarding the results of the different hierarchies that produced the feature vectors on Chapter 4, there exist very little differences that could be perceived if we showed all the decimal of the results, but nevertheless, we can state that the three hierarchies are equally good, since no one stands out over the others. We had hoped that at least the hierarchy formed by the weekly profile would have a different result, since it was the most different, but this was not the case. In Figures 5.2 and 5.3 we can see that in models $minT_1$ and $minT_{24}$, which make use of the predictions of the intermediate levels, no relevant distinction appears between using one hierarchy or another.



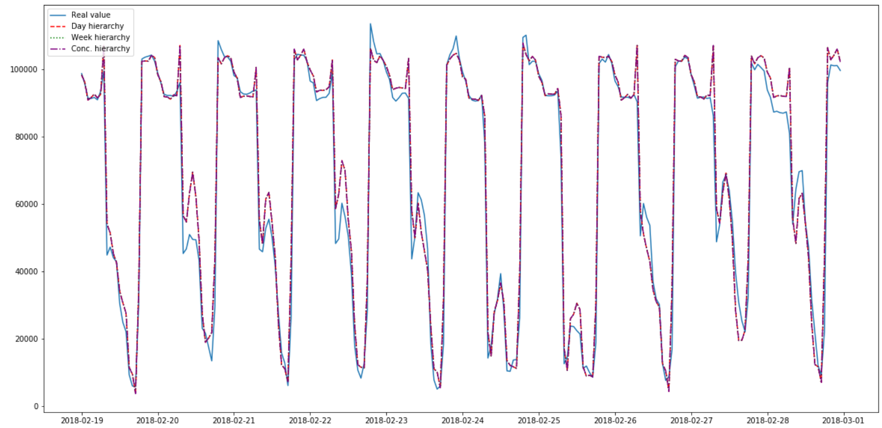
(a) $minT_1$ with OLS



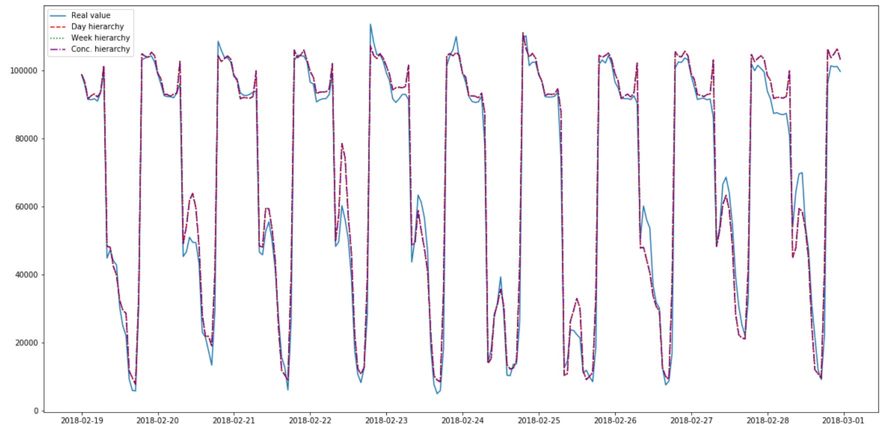
(b) $minT_1$ with WLSV

Figure 5.2: Winter comparison of the hierarchies with each reconciliation method

Concerning the ME in Table 5.3, almost all models tend to over-forecast except the models $root_1$ and $minT_1$ with OLS reconciliation, which under-forecast almost the same quantity. The first variant appears to be the most heterogeneous in that sense, since it presents very different values, having negative values, positive values and close to zero values, which expresses the all three models present distinct forecasting behaviours. This changes of conduct can be seen in Figure 5.4 in which we use only the best model of $minT_1$ (OLS) compared to the other two of the same variant. Another thing that we can observe is that the $root_1$ and $minT_1$ tend to under-forecast, whereas the model BU_1 is inclined to over-forecast.



(a) $minT_{24}$ with OLS



(b) $minT_{24}$ with WLSV

Figure 5.3: Winter comparison of the hierarchies with each reconciliation method

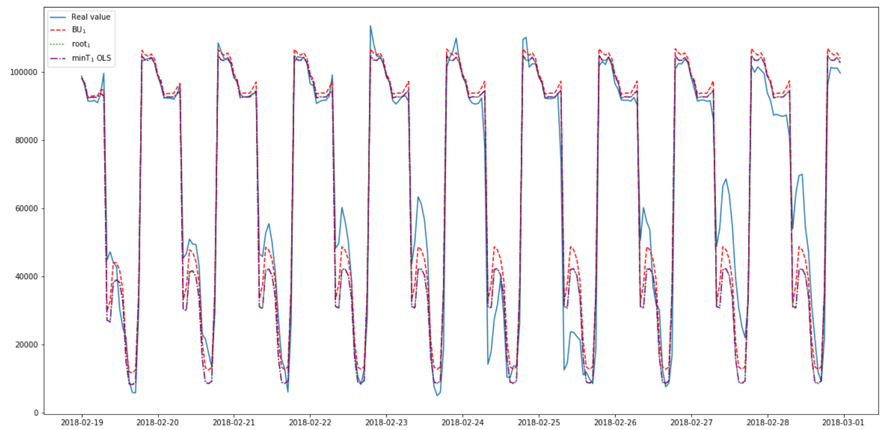


Figure 5.4: Winter comparison of the one predictor variant models

On the other hand, the second variant is quite homogeneous, showing values between the range of 1650 and 1900, that considering the scale of the data is can be said that they are similar. In Figure 5.5 we can observe that the deviations between models predictions are smaller than the models of the first variant. As with the comparison of the other variant, we only used the best model of $minT_{24}$ (WLSV) in the Figure.

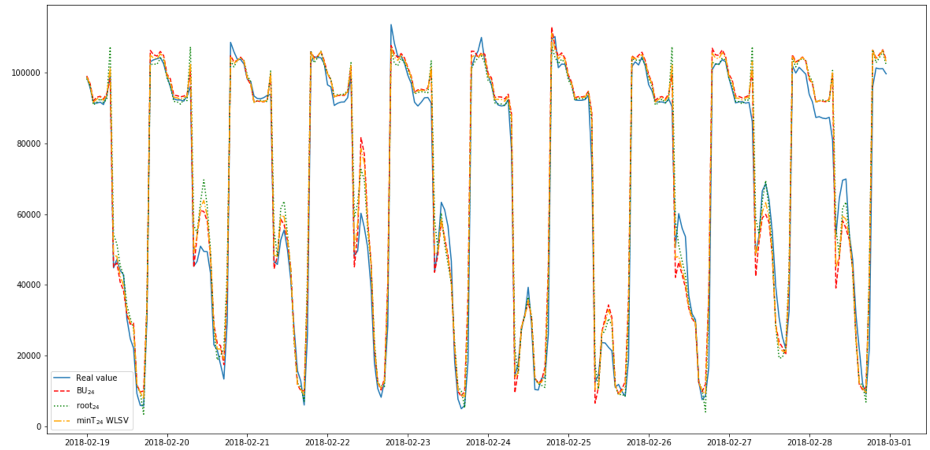


Figure 5.5: Winter comparison of the 24 predictors variant models

With respect to the MAE metric in Table 5.3, we are also able to find dissimilarities between both variants. The models with one predictor by node have more or less 6000 kWh of error while the models with 24 predictors by node decrease this number by 2000, thus giving rise to an average of 4000 kWh of error.

The MAPE results shown in the same Table (5.3) reveal that the predictor at the root node ($root_1$ and $root_{24}$) has more accuracy than the bottom-up model (BU_1 and BU_{24}) in both variants, but the reconciled models perform better, at least with one of the reconciliation methods, supporting our hypothesis that hierarchical forecasting reduces the error of the base predictions.

In regard to the reconciliation methods, and looking at the MAPE values of the Table (5.3), it appears that OLS performs better when we only have one predictor per node, and WLSV when we have 24 predictors per node. This may be due to the fact that WLSV estimates the matrix W_h, mentioned in section 5.1.2, from the matrix W_1 (the covariance matrix of the 1-step ahead base prediction errors), and therefore, since the models of 24 predictors have a step prediction of $h = 2$, we can expect W_2 to be very similar to W_1. On the other hand, for the one predictor models, we have a step prediction of 24 to 48 and the approximation based on W_1 struggles. In contrast, we would say that OLS does not function better with the models of one predictor per node, but that WLSV behaves worse. Figures 5.6 and 5.7 display that divergence on accuracy between both reconciliation methods.

In the last figure, Figure 5.8, we can perceive something interesting between the best model of each variant, that the results can not show. It appears that $minT_1$ with OLS tends to follow a more constant behaviour, however, $minT_{24}$ with WLSV tries to fit the variation of the actual values, which seems to be another reason to explain why the second variant presents better results.

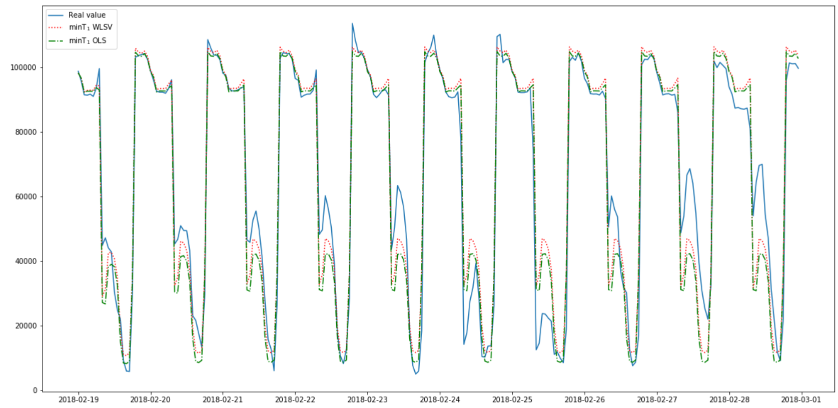


Figure 5.6: Winter comparison of the reconciliation methods in $minT_1$

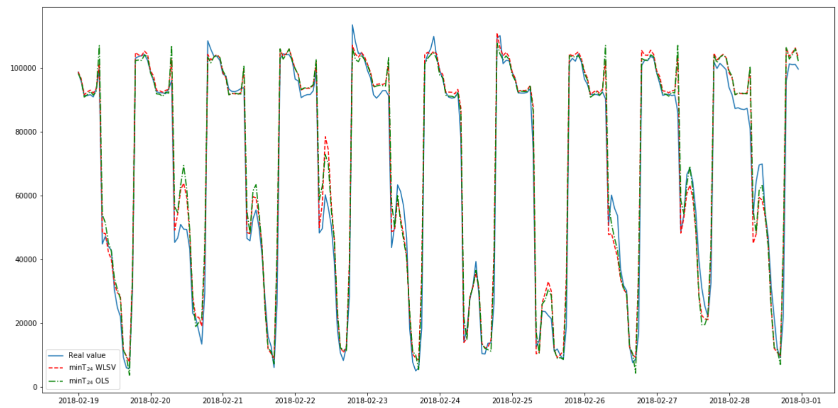


Figure 5.7: Winter comparison of the reconciliation methods in $minT_{24}$

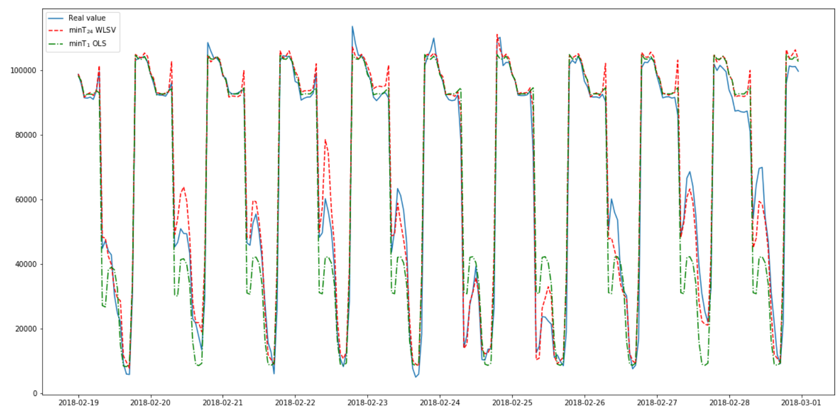


Figure 5.8: Winter comparison of the best reconciled model from $minT_1$ and $minT_{24}$

On the other hand, we also reasoned that it would be interesting to analyze the error of the predictions by time slot, in order to evaluate the performance of the predictors of the second variant. The results can be observed in table 5.4. In general, we can see that all models have a significant increase of the percentage of error at 7H that then grows slowly till 14H when another boost of the error occurs. After that, it appears another more rapid raise, and a huge boost of the error at 18H, the hour with the most percentage of error. Finally, the last hours (19H to 23H) present low MAPE values. This seems to support the idea that it is more difficult to predict the behaviour of the working hours, which can present a more diverse behaviour. In fact, the 18H often coincides with the end time of working hours, which may explain the peak of error.

Another thing that we can observe is that all three models present very similar MAPE results considering the scale of error in each time slot, with only a few hours in which we can detect strange behaviours, like at 8H where there is a significant reduction in BU_{24} and a greater one in $minT_{24}$ with WLSV.

It is relevant to mention, that in this table (5.4), the hierarchies also do not show up any important difference. Also, it is of interest that examples can be found in which each type of model is better than the rest, for example, in 7H, the best model is BU_{24}; in 10H, the best is $root_{24}$; in 0H, the model $minT_{24}$ with WLSV predominates; and lastly, in 19H, the best model is $minT_{24}$ with OLS. This implies that, although in general, the best results for this variant with the dataset of WINPRED01 are the ones coming from $minT_{24}$ with WLSV, by time slots, it does not hold always true.

Model	BU_{24}	$root_{24}$	$minT_{24}$					
Reconciliation	None	None	OLS			WLSV		
Hierarchy	Not relevant	Not used	Day	Week	Conc.	Day	Week	Conc.
0H	1.61	1.71	1.69	1.69	1.69	1.45	1.45	1.45
1H	2.22	1.45	1.47	1.47	1.47	1.9	1.9	1.9
2H	1.82	1.62	1.61	1.61	1.61	1.53	1.53	1.53
3H	2.32	1.48	1.49	1.49	1.49	1.98	1.98	1.98
4H	2.19	1.69	1.7	1.7	1.7	1.88	1.88	1.88
5H	2.06	1.36	1.37	1.37	1.37	1.81	1.81	1.81
6H	1.96	1.48	1.47	1.47	1.47	1.62	1.62	1.62
7H	10.67	13.74	13.66	13.66	13.66	11.57	11.57	11.57
8H	15	25.64	24.57	24.57	24.57	7.77	7.77	7.77
9H	12.54	14.46	14.38	14.38	14.38	12.88	12.88	12.88
10H	14.86	11.29	11.35	11.35	11.35	13.4	13.4	13.4
11H	17.65	14.31	14.33	14.33	14.33	16.4	16.4	16.4
12H	14.26	12.94	12.98	12.98	12.98	13.8	13.8	13.8
13H	13.75	13.59	13.6	13.6	13.6	13.7	13.7	13.7
14H	19.51	20.66	20.63	20.63	20.63	19.91	19.91	19.91
15H	21.55	24.04	23.93	23.93	23.93	21.5	21.5	21.5
16H	27.54	31.5	31.36	31.36	31.36	28.36	28.36	28.36
17H	38.25	24.53	23.55	23.55	23.55	22.09	22.09	22.09
18H	63.46	59	59.12	59.12	59.12	61.91	61.91	61.91
19H	4.53	3.39	3.36	3.36	3.36	3.95	3.95	3.95
20H	2.03	2.24	2.21	2.21	2.21	1.94	1.94	1.94
21H	1.78	1.35	1.36	1.36	1.36	1.53	1.53	1.53
22H	2.48	1.83	1.83	1.83	1.83	2.16	2.16	2.16
23H	1.97	1.06	1.07	1.07	1.07	1.59	1.59	1.59

Table 5.4: Winter MAPE results in percentage for each time slot

5.3.2. Summer season results

The statistics of the models with the dataset of SUMPRED02 can be seen in Table 5.5. Contrary to what happened with the winter dataset, we cannot state that the models that have one predictor per node (BU_1, $root_1$, $minT_1$) have less accuracy than the models that present 24 predictors (BU_{24}, $root_{24}$, $minT_{24}$), considering the MAPE metric, although in average, the 24-hour variant models have more accuracy. In fact, the best prediction model would be $root_1$, giving us the insight that with the summer season a hierarchical forecasting model is not needed.

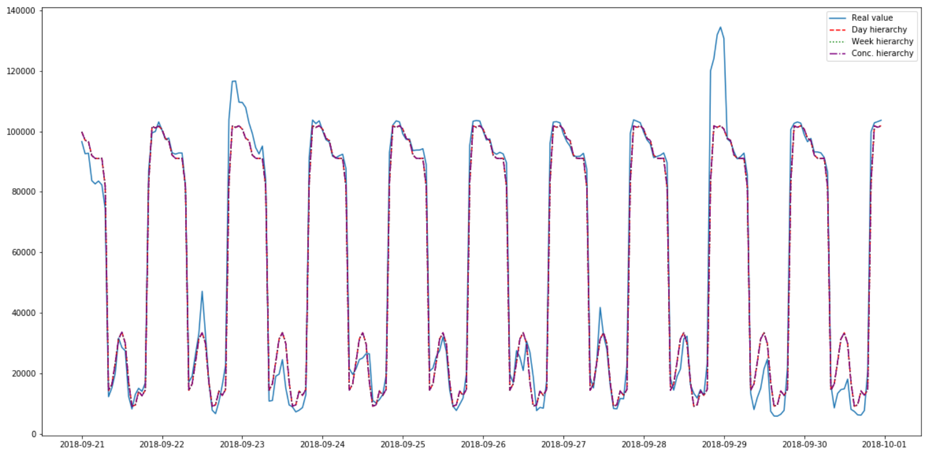
Model	Reconciliation	Hierarchy	ME	MAE	MAPE
BU_1	None	Not relevant	1263.6765	4797.5865	0.2470
$root_1$	None	Not used	-915.3680	4610.1531	0.1830
$minT_1$	OLS	Day	-858.0247	4600.9833	0.1840
		Week	-858.0247	4600.9833	0.1840
		Conc.	-858.0247	4600.9833	0.1840
	WLSV	Day	292.3206	4567.8879	0.2130
		Week	292.3206	4567.8879	0.2130
		Conc.	292.3206	4567.8879	0.2130
BU_{24}	None	Not relevant	1077.2804	4180.2191	0.1865
$root_{24}$	None	Not used	-392.1624	4798.1969	0.1952
$minT_{24}$	OLS	Day	-353.4929	4772.8913	0.1947
		Week	-353.4929	4772.8913	0.1947
		Conc.	-353.4929	4772.8913	0.1947
	WLSV	Day	423.1194	4375.5318	0.1871
		Week	423.1194	4375.5318	0.1871
		Conc.	423.1194	4375.5318	0.1871

Table 5.5: Summer results

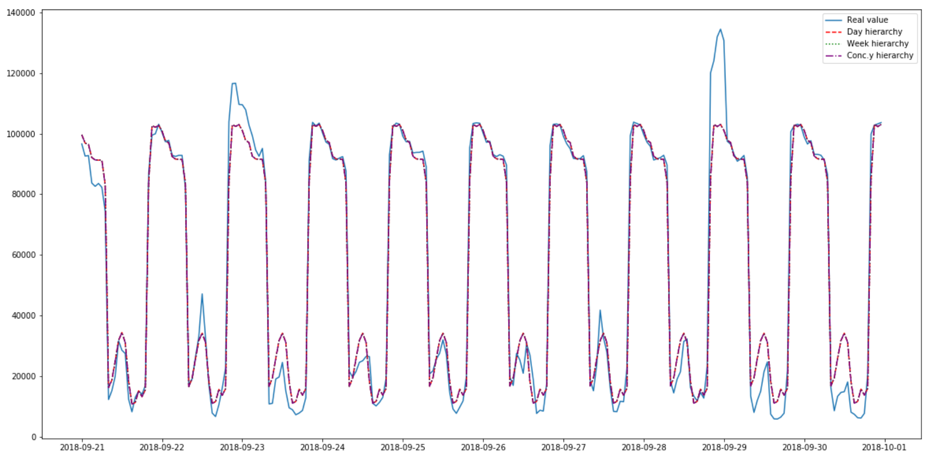
When it comes to the different hierarchies already commented, as with the winter season, we can state that all three configurations of the hierarchy are to the same degree good. In Figures 5.9 and 5.10 we can see that in models $minT_1$ and $minT_{24}$, there is not any important difference between using one hierarchy or another.

With respect to the ME metric in Table 5.5, it appears that there is a more varied behaviour. We can notice that *root* and *minT* with OLS models of both variants tend to under-forecast more than what they over-forecast, meanwhile *BU* and *minT* with WLSV tend to the contrary. Also, since the values of the ME are smaller, it indicates that the behaviour of the model does not have a clear tendency to over or uder forecast. This behaviours of the models can be seen in Figures 5.11 and 5.12.

Regarding the MAE metric on the same Table (5.5), we can spot an homogeneous set of values in both variants, presenting an average of 4600 kWh of error. Although, it is relevant to mention that the models with 24 predictors show the widest range of values, since the model $root_{24}$ is the one with the greatest value and the model BU_{24} the one with the smallest.



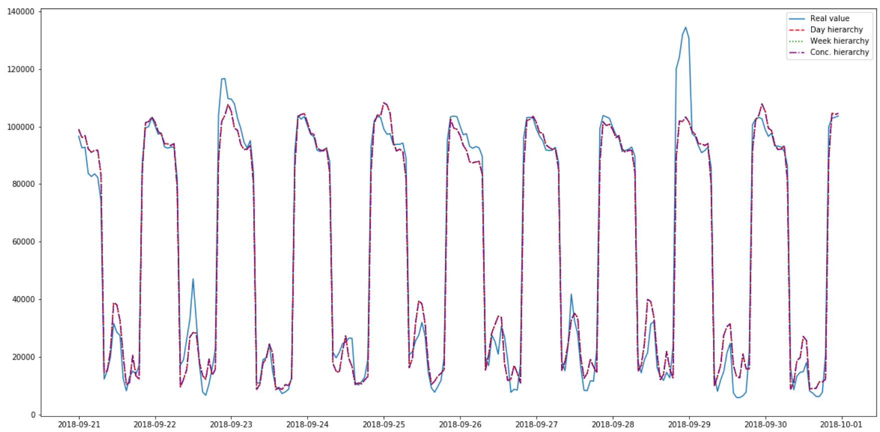
(a) $minT_1$ with OLS



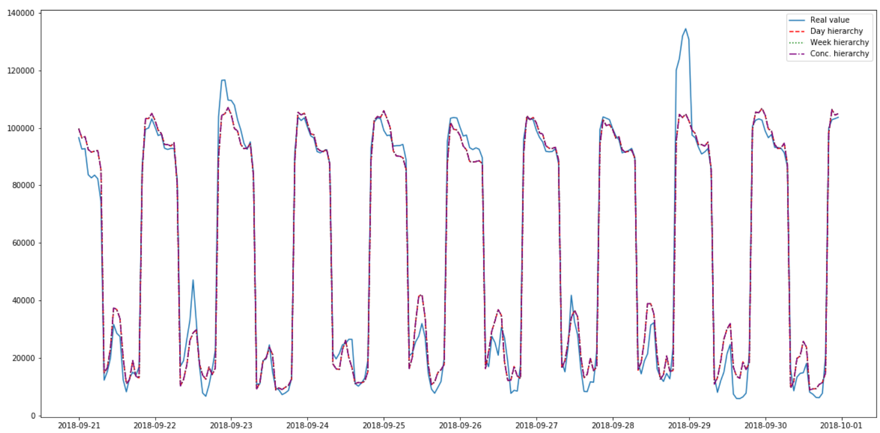
(b) $minT_1$ with WLSV

Figure 5.9: Summer comparison of the hierarchies with each reconciliation method

In connection with the MAPE results, that we can see in Table 5.5, they reveal that there is in average of 20% of error and that the increase of accuracy appears not to be affected by the number of predictors per node. On the other hand, contrary to what we saw in winter results, the root node ($root_1$ and $root_{24}$) does not always have more accuracy than the bottom-up model (BU_1 and BU_{24}), and the reconciled models also do not seem to perform better, which contradicts our hypothesis that hierarchical forecasting reduces the error of the base predictions, at least with the summer season. It is relevant to mention, that it does not appear any relevant decrease in the percentage of error, other than that seen in models BU_1 and $root_1$, which present a reduction of 6% of error.



(a) $minT_{24}$ with OLS



(b) $minT_{24}$ with WLSV

Figure 5.10: Summer comparison of the hierarchies with each reconciliation method

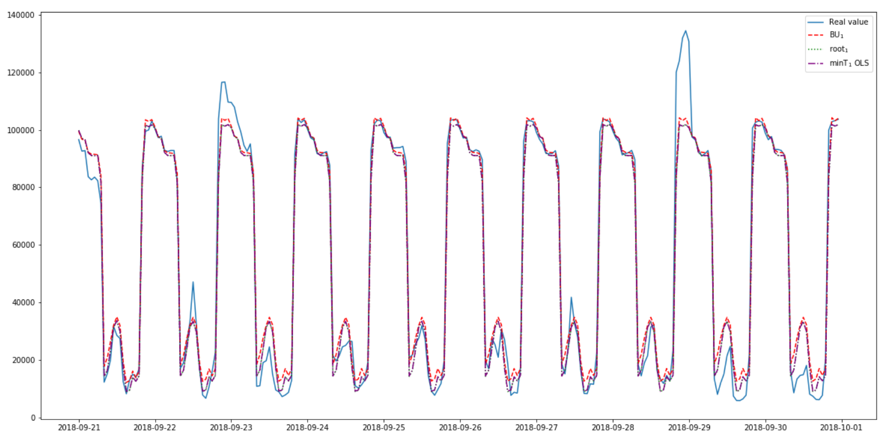


Figure 5.11: Summer comparison of the one predictor variant models

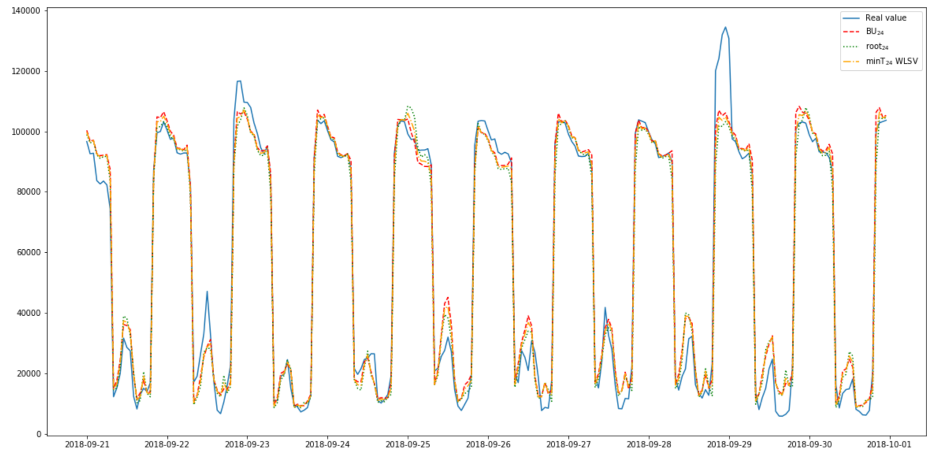


Figure 5.12: Summer comparison of the 24 predictors variant models

In regard to the performance of the reconciliation methods, considering the MAPE metric in Table (5.5), we find similar analysis to the ones of the winter results. The same reason for which OLS function better with one predictor models and WLSV with 24 predictors looks like to be maintained. This is supported by the fact that OLS stills performs better in the model of the first variant (1 predictor) and WLSV in the model of the second one (24 predictors). In Figures 5.13 and 5.14 we can also see the difference between both reconciliation methods.

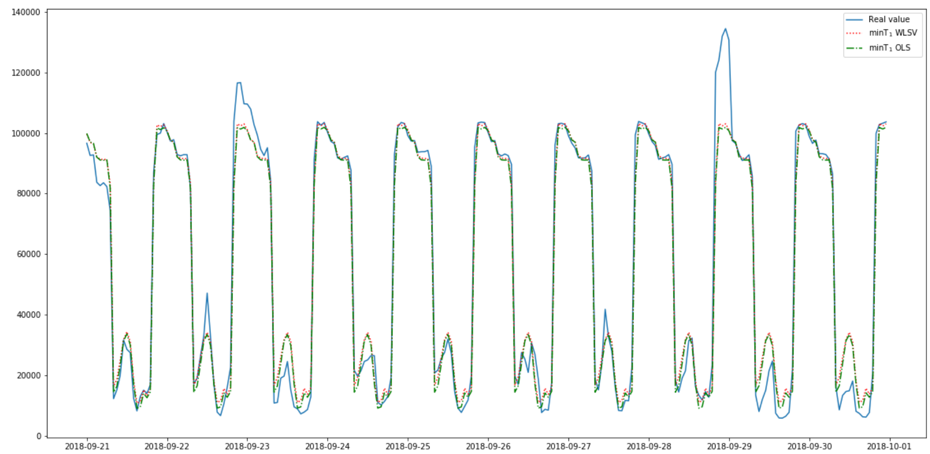


Figure 5.13: Summer comparison of the reconciliation methods in $minT_1$

Last figure, Figure 5.15, reveals again what we have recognized from the table 5.5, that the models show very little differences between them, since we only can perceive little differences between the time series of the predictions.

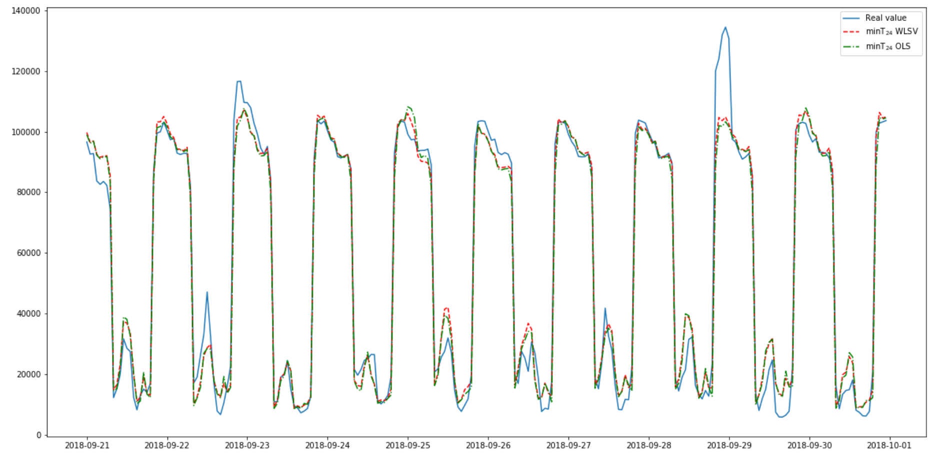


Figure 5.14: Summer comparison of the reconciliation methods in $minT_{24}$

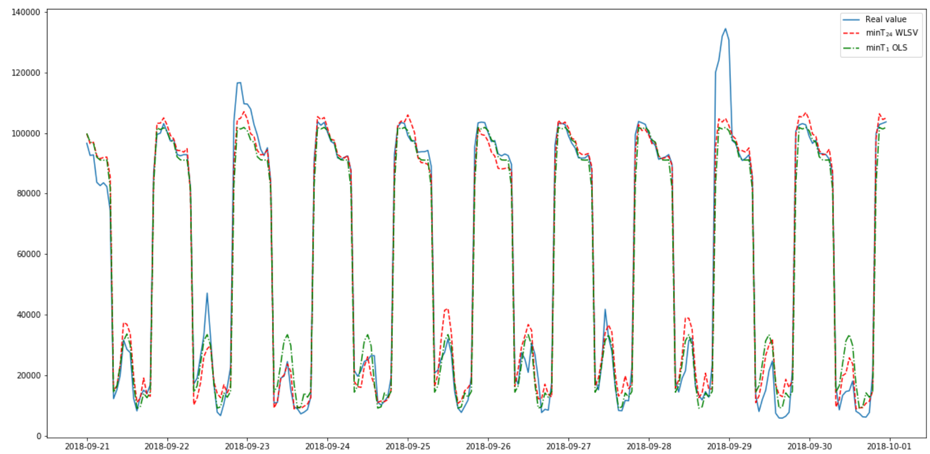


Figure 5.15: Summer comparison of the best reconciled model from $minT_1$ and $minT_{24}$

In addition to this, and as we did in the results of the winter dataset, WINPRED01, we have also evaluated the percentage of error by time slots. The results can be observed in table 5.6. In general, we can see that all models follow a a behavior similar to that already explained in the winter counterpart, but with an increase in the error in all time slots, since, as we have mentioned throughout the analysis, it seems to be more difficult to predict in summer. Also, this behaviour has a shift of one hour, starting the increase of error one hour later, at 8H instead of 7H, this may be due to daylight saving hours, in fact, in march we change one hour forward, which fits perfectly with what we have seen.

As with season of winter, we can observe very similar MAPE results in each time slot, with only a few strange behaviours in particular hours. The most eye-catching variations between models can be found at 19H and 20H where BU_{24} surprises with a huge decrease of the error.

Finally, in this table (5.6) as with the table 5.4, the hierarchies also do not indicate that there exist differences between them, and examples of time slots in which different models performs better than the rest appear. For example, at 0H, BU_{24} has the best results, and at 1H, $root_{24}$ takes the advantage.

Model	BU_{24}	$root_{24}$	$minT_{24}$					
Reconciliation	None	None	OLS			WLSV		
Hierarchy	Not relevant	Not used	Day	Week	Conc.	Day	Week	Conc.
0H	4.98	5.28	5.27	5.27	5.27	5.11	5.11	5.11
1H	2.97	3.34	3.33	3.33	3.33	3.13	3.13	3.13
2H	2.41	2.87	2.85	2.85	2.85	2.44	2.44	2.44
3H	3.36	2.91	2.9	2.9	2.9	3.06	3.06	3.06
4H	3.05	2.97	2.97	2.97	2.97	2.95	2.95	2.95
5H	2.84	2.22	2.22	2.22	2.22	2.44	2.44	2.44
6H	3.47	2.83	2.84	2.84	2.84	3.14	3.14	3.14
7H	4.71	6.54	6.34	6.34	6.34	2.78	2.78	2.78
8H	18.83	25.32	25.15	25.15	25.15	21.72	21.72	21.72
9H	28.65	22.92	22.94	22.94	22.948	25.53	25.53	25.53
10H	28.9	23.07	23.2	23.2	23.2	25.77	25.77	25.77
11H	34.75	34.58	34.56	34.56	34.56	34.5	34.5	34.5
12H	33.77	32.1	32.15	32.15	32.15	33.03	33.03	33.03
13H	23.18	21.94	21.97	21.97	21.97	22.63	22.63	22.63
14H	34.03	35.94	35.89	35.89	35.89	34.88	34.88	34.88
15H	41.77	36.06	36.19	36.19	36.19	36.69	36.69	36.69
16H	52.02	47.76	47.85	47.85	47.85	49.63	49.63	49.63
17H	57.29	70.52	70.15	70.15	70.15	62.69	62.69	62.69
18H	34.4	34.63	34.63	34.63	34.63	34.51	34.51	34.51
19H	13.69	30.8	30.34	30.34	30.34	21.15	21.15	21.15
20H	5.12	11.36	11.13	11.13	11.13	6.57	6.57	6.57
21H	4.8	3.98	3.93	3.93	3.93	4.17	4.17	4.17
22H	4.73	4.7	4.7	4.7	4.7	4.72	4.72	4.72
23H	4.06	3.94	3.94	3.94	3.94	4	4	4

Table 5.6: Summer MAPE results in percentage for each time slot

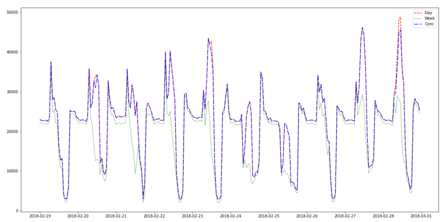
5.4 Conclusions

All things considered, after seeing the results on both datasets, the one from winter, WINPRED01, and the one from summer, SUMPRED02, we can conclude that our reasoning about a coherent hierarchical forecasting being more helpful to increase the accuracy of the base predictions holds only with the winter season as the best model for summer does not take into account the hierarchy. This difference of the behaviour from one dataset to another may be due to the variability of the observed values in each season. Winter tends to be a season with a more constant behaviour without big outliers (values very different of the other values), as the days are shorter and the temperatures are more steady. On the other hand, the season of summer seems more dependant of external factors like the temperature. Although the Mediterranean coast has more stable climates, it is true that heat waves appear intermittently in summer, which may cause abrupt changes in electricity consumption behavior, such as turning on the air conditioning or stopping the activities that were made at that moment. This rationale may be supported by the fact that the predictions of winter show more accuracy in general, and also a decreasing of the error in the models, meanwhile, the models in summer reveal a bigger and less different percentage of error. Furthermore, we can notice more spikes in the figures of the summer season.

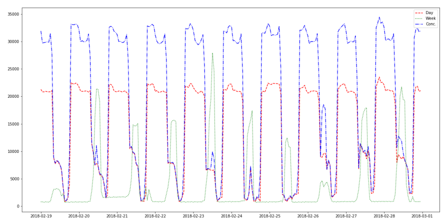
For the part of the different hierarchies constructed with the three feature vectors of Chapter 4, both results, demonstrate that the three hierarchies have such similar results that the differences cannot be seen in the figures and results without enlarging them to a greater degree of detail. Even the time series of the clusters, which can be seen in Figure 5.16, show significant differences in two of them. Nevertheless, it seems that by aggregating the four time series into the total aggregate series or root, it clears the differences producing what we have already commented, that they are extremely similar.

On the other hand, the result of the errors by time slot appear to support our first conclusion, in which summer is a season with more difficulties to forecast. Meanwhile, winter by presenting a more constant behaviour, improves the accuracy of the predictions. Furthermore, the analysis shows that the working hours which present more variability on its behaviour also present added challenges.

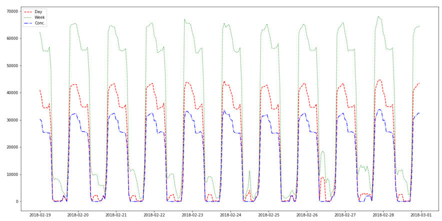
Finally, we have to mention that we also tried a model in which the reconciliation is made by a neural network, everything seemed to indicate that a neural network receiving as input the base predictions and outputting the reconciled predictions would make sense, since it would be able to find the contribution of each series to the reconciled one. However, the results did not show a consistent behavior and the best percentage errors associated with this model were still too high compared to those previously seen. For this reason, it was decided not to continue with this way.



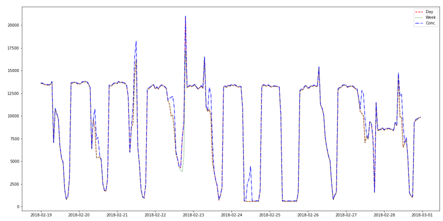
(a) Cluster 1



(b) Cluster 2



(c) Cluster 3



(d) Cluster 4

Figure 5.16: Comparison of the time series of each hierarchy by cluster

CHAPTER 6
Conclusion

In this project we have managed to accomplish the objectives that we proposed to ourselves in Chapter 1. After having gone through the entire process to build a model for predicting electricity consumption, we feel that we are prepared to synthesize the development of this work.

The first objective that we proposed was to construct a hierarchical structure that would group the load series by the similarities of data. This objective was achieved thanks to the knowledge acquired in the Intelligent Systems course, which was later reinforced in the Social Web Behaviour course of the fourth year, where we learned how to do this through clustering, in particular KMeans. It is true that the rest of the methods were unknown to us and it was necessary to investigate their details in order to analyze the results obtained correctly. It is also worth mentioning that Gaussian mixture models were seen in the Machine Learning course, although we did not initially realize that they could be used for this purpose. In general, we spent too much time on this objective, as the decision to focus on the winter and summer seasons was not made until much later. Therefore, the time spent on clustering for a whole year was mostly wasted, although it also helped us to know how to write the Chapter 4 correctly.

The second objective that was presented was to design and develop a load forecasting model to predict the total consumption of an energy consumer. This was the most uncertain objective of the project, because during the undergraduate course we do not see problems of regression, i.e. prediction of the next values in a series, and in particular the techniques and methods used. But we do not start from a total lack of knowledge, since in subjects such as Intelligent Systems, Perception and Machine Learning we see classification problems, which after all are prediction problems in which the class to which a sample belongs is predicted. Furthermore, they also provided us with the mathematical basis to tackle this project and the methodology for analyzing different models.

On the other hand, and continuing with this objective, we had an added difficulty since the results of the summer season did not comply with the hypothesis that we had proposed, in which the hierarchy made in the first objective was used to obtain better results in the prediction models, so we repeatedly asked ourselves if we were doing the right thing, investigating in turn other possible ways, like a neural network. In the end, the results presented in Chapter 5 are all we could explore after verifying that the results were what they were and that there were no errors in the implemented code.

Finally, we believe that this project can be expanded through the following avenues:

- Exploring exogenous variables: during this project we have tried different predictor models such as SARIMA. These models accept variables that depend on factors external to the data set. It would be interesting to investigate whether climatological data such as temperature, humidity, luminosity or whether there has been precipitations can help to improve the accuracy of the models by predicting behavioral changes in the time series.

- Testing other dataset: it is possible that with another data set with the same electricity consumption prediction problem, our hypotheses could be fulfilled in the two seasons of the year, winter and summer. Therefore, reinforcing the work and decisions made in this project.

- Changing the scale of the problem: this project included a small town of Valencia, Llíria, so we chose to use an approach in which the hierarchy was built by similarity between the data. It would be intriguing to investigate whether in problems of another scale, such as at the level of a building where we would have the consumption data of different appliances or at the level of an autonomous community where we would have the consumption data of its localities, this type of data-driven hierarchy would work as we would expect, thus, improving the predictions of the models.

- Investigating other approaches: another way to continue this work would be to try to implement other hierarchical forecasting approaches such as the GTOP algorithm presented in Chapter 2.

Bibliography

[1] Nazih Abu-Shikhah and Fawwaz Elkarmi. Medium-term electric load forecasting using singular value decomposition. *Energy*, 36(7):4259–4271, jul 2011. doi:10.1016/j.energy.2011.04.017.

[2] Nima Amjady and Farshid Keynia. Mid-term load forecasting of power systems by a new prediction method. *Energy Conversion and Management*, 49(10):2678–2687, oct 2008. doi:10.1016/j.enconman.2008.04.008.

[3] Timothy R. Anderson, Tugrul U. Daim, and Jisun Kim. Technology forecasting for wireless communication. *Technovation*, 28(9):602–614, sep 2008. doi:10.1016/j.technovation.2007.12.005.

[4] Pituk Bunnoon, Kusumal Chalermyanont, and Chusak Limsakul. Mid term load forecasting of the country using statistical methodology: Case study in thailand. In *2009 International Conference on Signal Processing Systems, ICSPS 2009*, pages 924–928, 2009. doi:10.1109/ICSPS.2009.174.

[5] Juan Pablo Carvallo, Peter H. Larsen, Alan H. Sanstad, and Charles A. Goldman. Long term load forecasting accuracy in electric utility integrated resource planning. *Energy Policy*, 119:410–422, aug 2018. doi:10.1016/j.enpol.2018.04.060.

[6] Damien Fay and John Ringwood. On the influence of weather forecast errors in short-term load forecasting models. *Power Systems, IEEE Transactions on*, 25:1751 – 1758, 09 2010. doi:10.1109/TPWRS.2009.2038704.

[7] Ümmühan Basaran Filik, Ömer Nezih Gerek, and Mehmet Kurban. A novel modeling approach for hourly forecasting of long-term electric energy demand. In *Energy Conversion and Management*, volume 52, pages 199–211. Elsevier Ltd, jan 2011. doi:10.1016/j.enconman.2010.06.059.

[8] David Fischer, Andreas Härtl, and Bernhard Wille-Haussmann. Model for electric load profiles with high time resolution for German households. *Energy and Buildings*, 92:170–179, apr 2015. doi:10.1016/j.enbuild.2015.01.058.

[9] Jack A. Goldstone, Robert H. Bates, David L. Epstein, Ted Robert Gurr, Michael B. Lustik, Monty G. Marshall, Jay Ulfelder, and Mark Woodward. A Global Model for Forecasting Political Instability. *American Journal of Political Science*, 54(1):190–208, jan 2010. URL: http://doi.wiley.com/10.1111/j.1540-5907.2009.00426.x, doi:10.1111/j.1540-5907.2009.00426.x.

[10] Luis Hernández, Carlos Baladrón, Javier Aguiar, Lorena Calavia, Belén Carro, Antonio Sánchez-Esguevillas, Francisco Pérez, Ángel Fernández, and Jaime Lloret. Artificial Neural Network for Short-Term Load Forecasting in Distribution Systems. *Energies*, 7(3):1576–1598, mar 2014. URL: http://www.mdpi.com/1996-1073/7/3/1576, doi:10.3390/en7031576.

[11] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32:914–938, 2016.

[12] Tao Hong, Pierre Pinson, Shu Fan, Hamidreza Zareipour, Alberto Troccoli, and Rob J. Hyndman. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond, jul 2016. doi:10.1016/j.ijforecast.2016.02.001.

[13] Rob J. Hyndman, Alan J. Lee, and Earo Wang. Fast computation of reconciled forecasts for hierarchical and grouped time series. *Computational Statistics and Data Analysis*, 97:16–32, may 2016. doi:10.1016/j.csda.2015.11.007.

[14] Irena Komprej and Peter Zunko. Short term load forecasting. In *Melecon*, pages 1470–1473. Publ by IEEE, 1991. doi:10.1007/978-1-4613-1073-0_9.

[15] Corentin Kuster, Yacine Rezgui, and Monjur Mourshed. Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, 35:257–270, 2017. URL: https://www.sciencedirect.com/science/article/pii/S2210670717305899, doi:https://doi.org/10.1016/j.scs.2017.08.009.

[16] K. B. Lindberg, P. Seljom, H. Madsen, D. Fischer, and M. Korpås. Long-term electricity load forecasting: Current and future trends. *Utilities Policy*, 58:102–119, jun 2019. doi:10.1016/j.jup.2019.04.001.

[17] Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham. An ensemble of neural networks for weather forecasting. *Neural Computing and Applications*, 13(2):112–122, may 2004. URL: https://link.springer.com/article/10.1007/s00521-004-0413-4, doi:10.1007/s00521-004-0413-4.

[18] Joaquim Massana, Carles Pous, Llorenç Burgas, Joaquim Melendez, and Joan Colomer. Short-term load forecasting in a non-residential building contrasting models and attributes. *Energy and Buildings*, 92:322–330, apr 2015. doi:10.1016/j.enbuild.2015.02.007.

[19] Guy R. Newsham and Benjamin J. Birt. Building-level occupancy data to improve ARIMA-based electricity use forecasts. In *BuildSys'10 - Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, pages 13–18. ACM Press, 2010. URL: http://portal.acm.org/citation.cfm?doid=1878431.1878435, doi:10.1145/1878431.1878435.

[20] Henri Nyberg. Dynamic probit models and financial variables in recession forecasting. *Journal of Forecasting*, 29(1-2):215–230, jan 2010. URL: http://doi.wiley.com/10.1002/for.1161, doi:10.1002/for.1161.

[21] Lior Rokach and Oded Maimon. Clustering methods. In Oded Maimon and Lior Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 321–352. Springer US, 2005. doi:10.1007/0-387-25465-X_15.

[22] Heng Shi, Minghao Xu, and Ran Li. Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN. *IEEE Transactions on Smart Grid*, 9(5):5271–5280, sep 2018. doi:10.1109/TSG.2017.2686012.

[23] Souhaib Ben Taieb, James W. Taylor, and Rob J. Hyndman. Coherent probabilistic forecasts for hierarchical time series. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3348–3357, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[24] Jingrui Xie Tao Hong and Jonathan Black. Global energy forecasting competition 2017: Hierarchical probabilistic load forecasting. *International Journal of Forecasting*, 35:1389–1399, 2019.

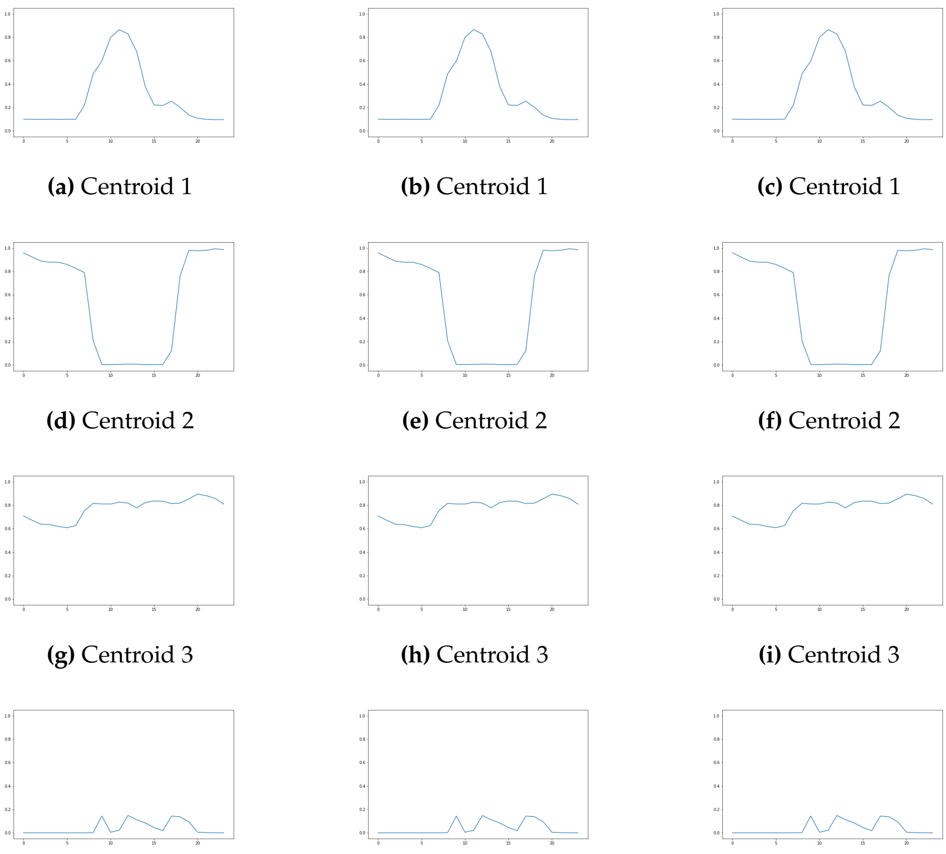
[25] Tim van Erven and Jairo Cugliari. Game-Theoretically Optimal reconciliation of contemporaneous hierarchical time series forecasts. In *Lecture Notes in Statistics*, volume 217, pages 297–317. Springer Science and Business Media, LLC, 2015. doi:10.1007/978-3-319-18732-7_15.

[26] Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019. doi:10.1080/01621459.2018.1448825.

[27] Yusheng Zhai. *Time series forecasting competition among three sophisticated paradigms*. PhD thesis, University of North Carolina at Wilmington, 2005.

APPENDIX A

Figures of the concatenated profile



KMeans | Gaussian Mixture | Mean Shift

(a) Centroid 1 (b) Centroid 1 (c) Centroid 1

(d) Centroid 2 (e) Centroid 2 (f) Centroid 2

(g) Centroid 3 (h) Centroid 3 (i) Centroid 3

(j) Centroid 4 (k) Centroid 4 (l) Centroid 4

Figure A.1: Winter Concatenated Profile clustering and centroids of the hourly part

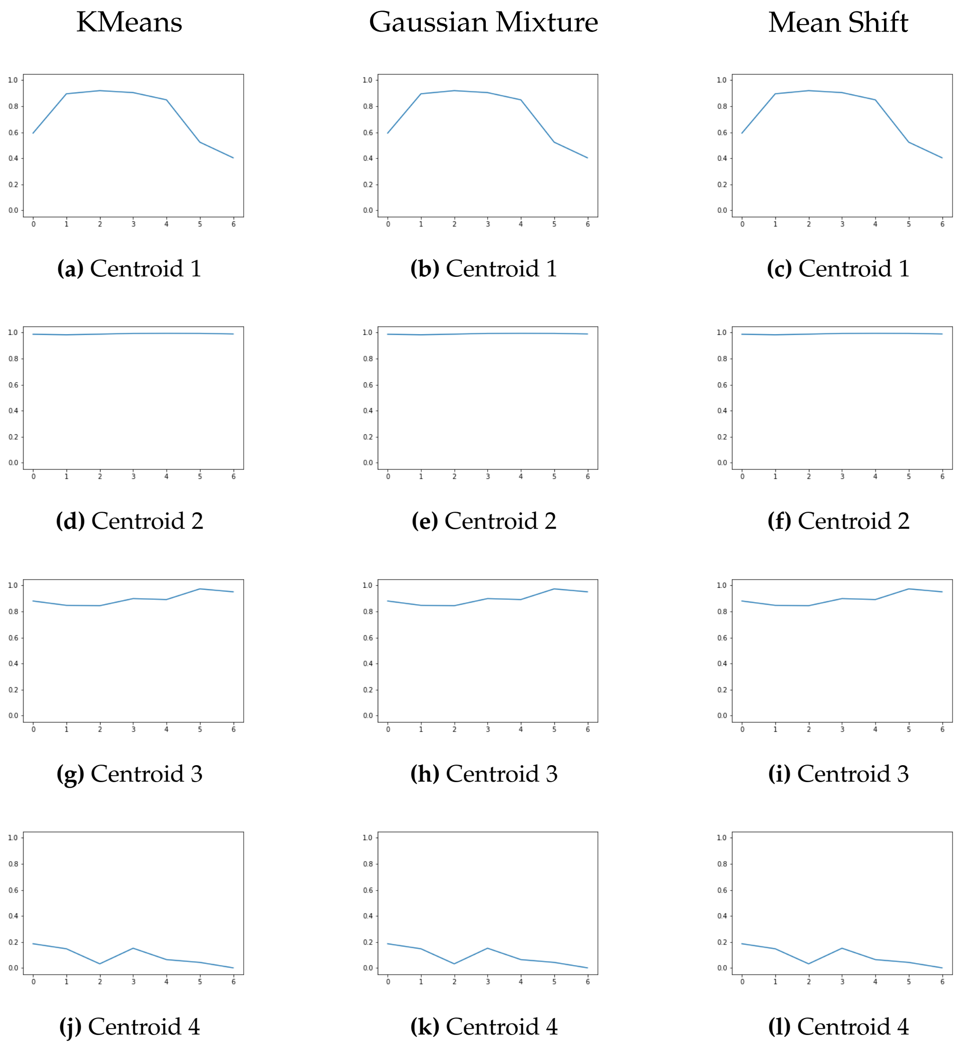
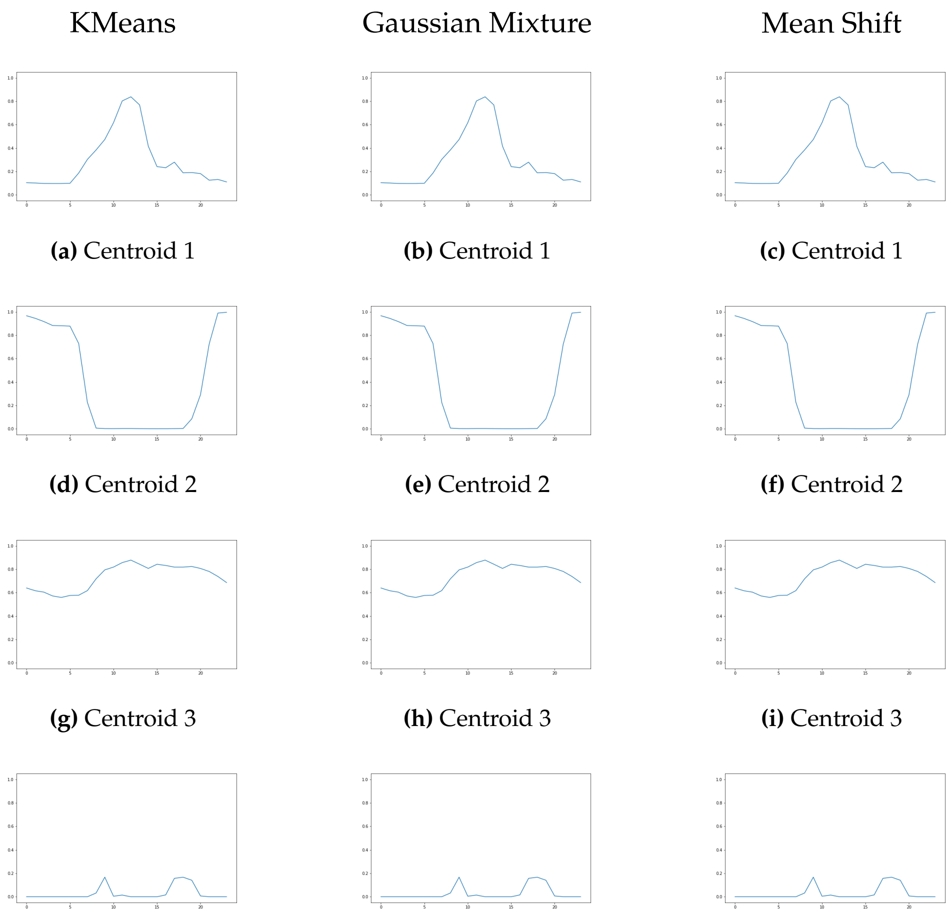


Figure A.2: Winter Concatenated Profile clustering and centroids of the daily part



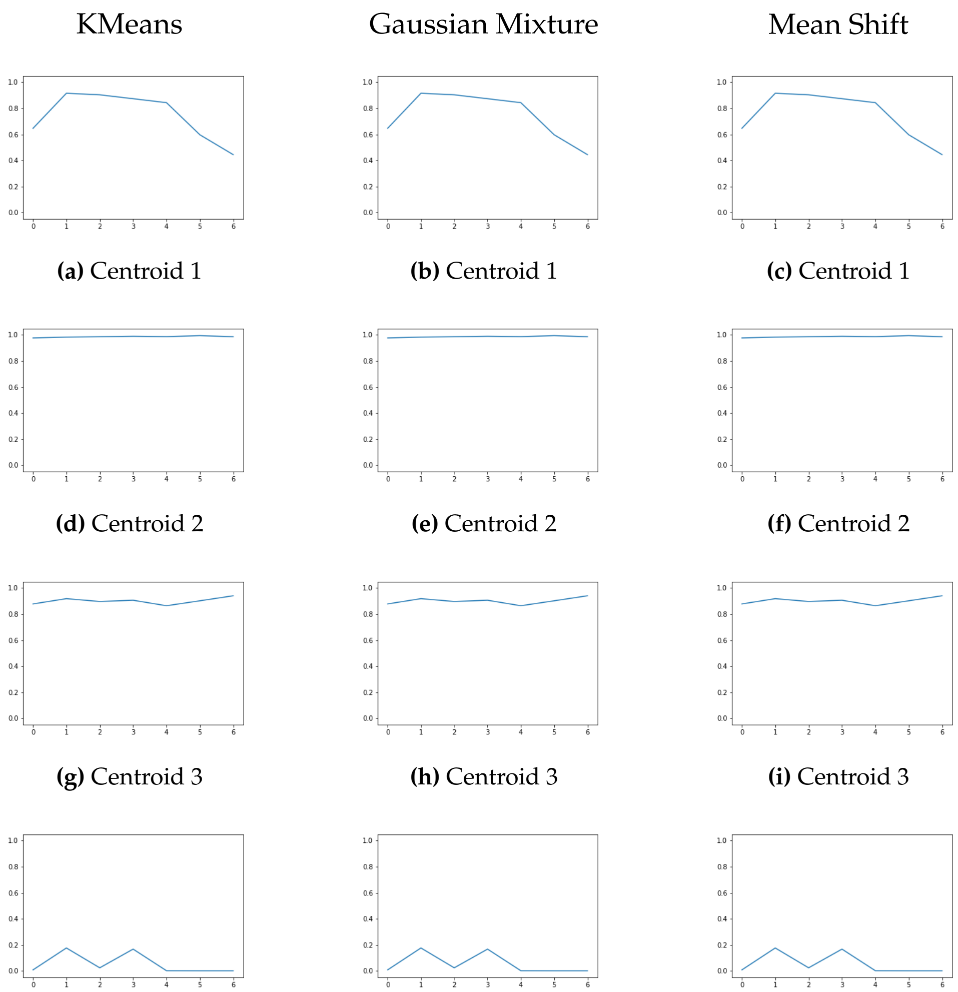
(a) Centroid 1 **(b)** Centroid 1 **(c)** Centroid 1

(d) Centroid 2 **(e)** Centroid 2 **(f)** Centroid 2

(g) Centroid 3 **(h)** Centroid 3 **(i)** Centroid 3

(j) Centroid 4 **(k)** Centroid 4 **(l)** Centroid 4

Figure A.3: Summer Concatenated Profile clustering and centroids of the hourly part



(a) Centroid 1 (b) Centroid 1 (c) Centroid 1

(d) Centroid 2 (e) Centroid 2 (f) Centroid 2

(g) Centroid 3 (h) Centroid 3 (i) Centroid 3

(j) Centroid 4 (k) Centroid 4 (l) Centroid 4

Figure A.4: Summer Concatenated Profile clustering and centroids of the daily part