UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Short-Term Electric Load Forecasting

Marco André Gonçalves Pinheiro

Supervisor: Doctor Sara Alexandra Cordeiro Madeira
Co-Supervisor: Doctor Alexandre Paulo Lourenço Francisco

Thesis approved in public session to obtain the PhD Degree in

Computer Science and Engineering

Jury final classification: **Pass with Distinction**

2023

TÉCNICO
LISBOA

UNIVERSIDADE DE LISBOA

INSTITUTO SUPERIOR TÉCNICO

Short-Term Electric Load Forecasting

Marco André Gonçalves Pinheiro

Supervisor: Doctor Sara Alexandra Cordeiro Madeira
Co-Supervisor: Doctor Alexandre Paulo Lourenço Francisco

Thesis approved in public session to obtain the PhD Degree in
Computer Science and Engineering

Jury final classification: **Pass with Distinction**

Jury

Chairperson: Doctor José Carlos Alves Pereira Monteiro
Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor Maribel Yasmina Campos Alves Santos
Escola de Engenharia, Universidade do Minho

Doctor Pedro Manuel Santos de Carvalho
Instituto Superior Técnico, Universidade de Lisboa

Doctor Alípio Mário Guedes Jorge
Faculdade de Ciências, Universidade do Porto

Doctor Alexandre Paulo Lourenço Francisco
Instituto Superior Técnico, Universidade de Lisboa

2023

Abstract

Energy forecasting covers a wide range of prediction challenges in the utility industry, such as forecasting demand, generation, price, and power load over diverse time horizons and at different levels of the power grid. The short-term load forecasting in low voltage, other than at the smart meters level, has not yet been carried out in-depth compared to national/regional/building load forecasting. It is then proposed a systematic approach from the system level to the low voltage considering not only the performance of the models but also their applicability, interpretability, and reproducibility. Considering an initial benchmark model, this is compared to improved GAM (generalized additive models) enhanced by introducing new explanatory variables, reducing error by 42-47% and preserving interpretability. Additionally, an ensemble method improves accuracy for specific periods in which modeling is particularly demanding using standalone GAM models. The method is applied to the national power load and, for the first time, to all 100 000 secondary substations that integrate the Portugal power grid, rather than to tackle the few open datasets. Additionally, an appropriate data representation of power load time series, transforming them into discrete symbol sequences, is proposed and used as the base to split similar load patterns within the year, week, and special days, forming clusters. Cluster-based models are then trained with stratified sampling from the respective cluster data using the same model structure. Individual models, cluster-based models, and one-size-fits-all model are compared in terms of accuracy and applicability. This approach is used to build a live daily forecasting system called PREDIS for the Portuguese DSO (Distribution System Operator) whose results anticipate load peaks and network constraints. It uses a distributed system architecture which copes with both capacity and scalability challenges inherent to the storage and processing of hundreds of thousands of time series and artifacts.

Keywords
Power load forecasting, secondary substations load, power load classification, cluster-based forecasting, train and inference distributed computing

Resumo

A previsão de energia abrange uma ampla gama de desafios de previsão no sector de energia, como previsão de demanda, geração, preço e carga em diversos horizontes temporais e em diferentes níveis da rede elétrica. A previsão de carga de curto prazo em baixa tensão, além do nível do consumidor final, não tem sido tão explorada em comparação com previsão a nível nacional/regional/edifício. Propõe-se então uma abordagem sistemática desde o nível do sistema até à baixa tensão considerando o desempenho dos modelos, bem como a sua aplicabilidade, interpretabilidade e reprodutibilidade. Considerando o modelo de benchmark inicial, este é comparado com modelos GAM (modelos aditivos generalizados) aprimorados pela introdução de novas variáveis explicativas reduzindo o erro em 42-47% e preservando a interpretabilidade. Adicionalmente, a combinação de diferentes modelos melhora a precisão em períodos específicos particularmente exigentes usando apenas modelos GAM independentes. O modelo é aplicado à carga elétrica nacional e, pela primeira vez, a todas as 100 000 subestações secundárias que integram a rede elétrica de Portugal, em vez de abordar os poucos conjuntos de dados abertos. Adicionalmente, uma representação de dados apropriada para séries temporais de carga de energia, transformando-as em sequências de símbolos, é proposta e serve de base para agrupar séries temporais com padrões de carga similares. Modelos baseados nesses clusters são treinados com amostra estratificada dos respetivos dados do cluster usando a mesma estrutura de modelo GAM, e comparando-os com os modelos individuais e com um modelo único global. Esta abordagem serve a construção de um sistema de previsão diária chamado PREDIS para o operador de distribuição de eletricidade português cujos resultados antecipam picos de carga e restrições de rede. Usa uma arquitetura de sistema distribuída que lida com os desafios de capacidade e escalabilidade de armazenamento e computação inerentes ao processamento de centenas de milhares de séries temporais e artefactos.

Palavras-chave
Previsão de carga de energia, carga em postos de transformação, classificação tipo de carga, previsão baseada em clusters, computação distribuída para treino e inferência modelos

Contents

Chapter 1

Introduction

Business and government organizations have succeeded due to effective planning, budgeting, and forecasting. They are generally considered crucial components of a company's performance management.

The utility sector is not an exception. Effective planning and forecasting are essential to a sector that trades such an important product, energy, being heavily regulated when compared to other sectors and its policies and decisions being under public scrutiny. Thus, energy forecasting is an essential task for daily operations and strategic decisions.

Since the first days of electricity distribution, utilities forecast the electricity demand for the next hours, days, and months. At that time, when electricity was essentially used for public illumination, forecasting demand was as easy as counting the number of light bulbs and multiplying by the power they consume. Today, energy forecasting is definitely no longer an easy activity. Electricity is consumed promptly almost anywhere and at any time by pressing a switch. Moreover, the energy transition, as the cornerstone of the upcoming developments of the energy system, includes the complex interaction of multiple technologies, business models innovation, and the decline of established business models and technologies [1].

As the system becomes increasingly complex and at the center of the energy transition, various business needs of energy forecasting have been reinforced within utilities [2] to cope with the upcoming increasing challenges of complexity, connectivity, scale, and scope. The following sections discuss this subject and summarize the different types of energy forecasting and time horizons depending on its goals.

1.1 Business Needs for Energy Forecasting

Nowadays, forecasting is an important task in all energy utilities, and its application extends throughout the value chain: production, transmission, distribution, and retail. Indeed, it is common to see *trading* in the high-level organizational structures of electric utilities alongside other business platforms. And, operationally, the trading unit supports their decisions on the available data and forecasting models. The business needs for energy forecasting could be summarized as follows.

Non-dispatchable generation forecasting [3] Utilities need to make efforts to accommodate more green energy towards a 100% renewable vision. As renewables grow, through the deployment of more wind and solar power, its availability has become less dispatchable. Wind and solar energy cannot be turned on and off at will, they are intermittent and come and go depending on the weather and time of day, so they must be accommodated [4]. Moreover, wind power also suffers from curtailment due to grid constraints or due to the abundance of wind power for such a momentary low demand for electricity. So, forecasting wind and solar energy for the next hours and days supports effective planning to accommodate that energy and to bidding it in the gross energy market if applicable, while long-term forecasting to find the best sites to deploy wind and solar farms.

Demand forecasting [5] Utilities, which operate in the retail segment of the value chain, also need to forecast the demand of their client portfolio for trading purposes or to analyze the power demand of individual clients to elaborate business term sheets, propose specific tariffs, and offer energy optimization services with a satisfying business rationale.

Trading in the gross market [6] Whether a utility sells its own energy generation, purchases energy for its consumers, or both, it must forecast the price of energy and plan when the best time to buy or sell. When applicable, utilities can negotiate long-term bilateral contracts and adjust in the daily wholesale market, besides providing grid services such as capacity services, energy shifting, and fast-response ancillary services. Independently of the utility's own strategy, forecasting energy and grid services price is important for decision marking and strategy execution.

Transmission and distribution (T&D) planning [7] As a Distribution or Transmission System Operator (DSO or TSO), the utility must maintain and upgrade the grid to meet the growth of demand and improve reliability. Planning decisions also rely on forecasts that inform when, where, and how much the load and the number of customers will grow. For example, the need for a new power substation in the future could imply the need to secure the land to place it.

Operations and maintenance [8] Forecasting the power load for the following days supports several decisions in the operations department. For example, scheduling maintenance without actual interruption to electricity consumers or to guide operators to make switching and loading decisions.

Demand Side Management [9] Able to manage dispatchable demand as more assets and models are introduced to increase flexibility on the demand side. When the power grid is overloaded or when the generation mix is not so green, controlled demand could be reduced instantly to lighten the system load. Demand-side management aggregators, virtual power plants, and logical dispatch of decentralized energy resources (DER) are models to add capacity and flexibility to the grid [4]. Predicting local power load for the next few minutes and hours is part of systems that (will) orchestrate air conditioners, local batteries, electric vehicle smart chargers, and other DER whose charges can be put out of step, desynchronized, or even anticipate/shifted to another time. Predictive capabilities are used for further system-level and local optimizations in accordance with a challenging dynamic of load fluctuations, grid restrictions, and dispatchable capacity and flexibility.

Financial and general planning Long-term energy forecasting and energy scenario predictions contribute to the management of company performance, helping executives project revenues, plan acquisitions, approve budgets, plan human resources, develop new business advantaged by technology and innovation, and other general business decisions. For example, electric vehicles (EV) are considered an important factor in the decarbonization of a large part of transport, and the power system must be able to meet their increasing charging needs with renewable energy. Smart charging technology would provide greater flexibility to the network and prevent investments in new electrical infrastructure [10]. Furthermore, the battery on wheels has been considered an asset in vehicle-to-grid (V2G), vehicle-to-home (V2H), and other usage and business models. Another example is the technology to store heating and electricity during low demand and high (renewable) energy availability, through centralized or decentralized technologies such as capacitors, superconductors, flywheels, batteries, heating storage, compressed air, pumped hydro, and (green) hydrogen (or other power-to-gas) [4, 11]. These are examples of technologies and innovations that have a high impact on electrification and descarbonization in the following years, and their business rationale must be supported by data analysis and predictions of energy scenarios.

As described, energy forecasting emerges as a fundamental enabler to tackle various perspectives. As an important tool, the accuracy of the forecast translates into the financial performance of energy utilities. A conservative estimate is that a 1% reduction in forecast error for a 10 GW utility can save up to $1.6 million annually [12]. Another estimate is that a 0.1% improvement in forecasting in a midsize European utility can help reduce about $3 million in operating costs in imbalance markets [13] considering the ability to forecast not only at the system level, but also at primary and secondary substations and at each energy point of delivery or generation as well, enabling optimization models such as dynamic tariffs, demand response, and DER management to peak shaving.

1.2 Types of Energy Forecasting

There is no single forecast that can meet all the utility needs. Different business purposes introduce diverse specificities and approaches. Electricity forecasting can be classified by three aspects:

- The **object** which is being estimated, such as (wind, solar, wave energy, hydroelectric,...) generation, load, demand, and energy price, as described in Table 1.1.
- The **time horizon** [2], which defines how far in advance the model forecasts, classified into four categories, as outlined in Table 1.2; although the definitions of the time horizon differ by author, it determines both the update cycle and the relevance of the explanatory variables.
- The **aggregation level**, such as the international integrated power grid (like the European electrical network), the regional or national DSO power grid, the primary or secondary substation loads, the energy point of delivery (highly dependent on consumption type), and virtual power plant's or energy retail portfolios.

Table 1.3 associates business needs with the type of forecast and the time frame.

From the classification framework of energy forecasting, the broadness of its application in the field is evident. This thesis aims to focus on a particular type of forecasting, temporal horizon, and aggregation level: short-term load forecasting at two levels, system and secondary substations (the latter also known as the object of low voltage forecasting). The following sections clarify the concept of load forecasting and the factors that influence its behavior.

1.3 Short-Term Electric Load Forecasting

Load forecasting is the technique used by power utilities to predict the energy required to meet demand and supply equilibrium. In other words, it predicts the net power load at a

Table 1.1. The object of forecasting.

Forecasting object
Generation forecasting whose object being estimated is the energy produced by a specific power plant, for instance hydroelectric, thermal, wind, solar or marine. Besides, forecasting intermittent energy sources as wind and solar come up as more complex and thoughtful.
Load forecasting is used to forecast the system load or energy flow in a specific bus, asset, or node within the power grid. While the system load is the sum of all the individual demands at all the nodes of the power system, the power load in a grid node can represent either a single consumer (typically a high- or medium-voltage one), a set of low-voltage consumers on a street or neighbor, an electric bus in a primary or secondary power substation that groups several consumers and small producers, a city, or even a region. *Spatial load forecasting* is a subtype that aims to estimate the future locations and magnitudes of power load within a utility's territory.
Demand forecasting aims to estimate the consumption of a single consumer or retailer's portfolio of consumers potentially dispersed along the area of influence of the utility.
Energy price forecasting focus on predicting price changes and futures in wholesale electricity markets.

Table 1.2. The time horizon and the time pace at which estimations are updated.

		Horizon	**Update Cycle**
VST	very short-term[i]	1 day	≤ 1 hour
ST	short-term	2 weeks	1 day
MT	medium-term	3 years	1 month
LT	long-term	3 decades	1 year

[i] VSTLF stands for very short-term load forecasting. The same analogy applies to STLF, MTLF, and LTLF.

Table 1.3. Type of forecasting in accordance with business requirements and time horizon.

	VST	**ST**	**MT**	**LT**	**Object**
Energy selling and purchasing					
Non-dispatchable generation	×	×	×		Generation
Electricity consumption	×	×	×		Demand
Trading Energy	×	×	×		Price
T&D planning			×	×	Load
Operation and maintenance	×	×			Load
Demand side management	×	×			Generation and Demand
Financial planning			×	×	Generation, Demand and Price

specific point or asset within the power grid. If at secondary substations, it is also known by low-voltage load forecasting. Utility systems rely on load forecasting for maintenance, scheduling, power generation planning (centralized and distributed), load switching, safety evaluation, cost optimization, and general guarantee of continuous power supply [12, 14, 15].

The dynamics of power loads is intrinsically related to (i) human activity and behavior on a daily basis, and (ii) its magnitude is due to the economy, land use, electrical efficiency, and how much the economy and society are electrified. The former is more dynamic and varies according to calendar, weather, and events in general, whereas the latter changes much slower. In addition to changing the magnitude of the power load over the years, the patterns and curve shapes can also change considerably with the introduction of new technologies and energy business models, such as distributed solar generation, distributed energy storage, demand side management, penetration of electrical vehicles, and energy communities that mean local settlements and optimizations among prosumers in neighbor.

It turns out that predicting these aspects and how they evolve in the near future is important when forecasting. However, some are available and accurate in the following days, but are unreliable for further days. For example, weather forecasting is accurate for the next days but is unskill for more than two weeks, whereas electrification does not change in the following months, and its growth is predictable for the next years. Table 1.4 summarizes the available features, how long they remain unchangeable, and how accurate its predictions are for load forecasting purposes. Consequently, Table 1.5 shows the use of each feature in load forecasting for different temporal horizons and, consequently, update cycles.

Table 1.4. Steadiness, accuracy, and availability of explanatory variables.

	Steadiness	**Accurate**	**Inaccurate**	**Unskill**
Weather	1 hour	1 day	2 weeks	> 2 weeks
Economics	3 months	6 months	3 years	> 3 years
Land Use	1 year	2 years	5 years	> 5 years
Electrification	1 year	2 years	5 years	> 5 years
Electrical Efficiency	1 year	2 years	5 years	> 5 years
Calendar	decades[ii]	—	—	—

[ii] Calendar is quite stable, except for rare jurisdiction changes on public holidays or saving light summer time.

Table 1.5. The use of each explanatory variable depends upon horizon of load forecasting.

	Horizon	**Update Cycle**	**Calendar**	**Weather**	**Economics**	**Electrification, Land Use and Efficiency**
VSTLF	1 day	≤ 1 hour	Required	Optional	Optional	Optional
STLF	2 weeks	1 day	Required	Required	Optional	Optional
MTLF	3 years	1 month	Optional	Simulated	Required	Optional
LTLF	3 decades	1 year	Optional	Simulated[iii]	Simulated	Required

[iii] Load forecasting longer than 30 years may take into account climate projections instead of weather predictions.

In VSTLF, weather, economics, electrification, electrical efficiency, and land use are optional variables because they are all relatively stable over a short time span. They remain unchangeable, and so the load for the next minutes and hours is estimated using autoregressive techniques.

On the other hand, weather predictions and calendar features play a key role in STLF, because the power load is driven not only by the time and day being forecasted, but also by the atmospheric conditions.

For longer horizons, the weather forecast is unreliable, and therefore simulated climate projections based on historical weather data may be used in MTLF. In contrast, economic variables change over the mid-term period, and, since they affect power load, they are important for forecasting power load up to three years.

Finally, in LTLF the same rationale applies: for variables with unreliable prediction methods, are changed by simulated probabilistic projections of the same variables. Additionally, variables once steady for shorter periods must be taken into consideration for longer periods because of their effects on power load in the long term, such as electrification, land use, electrical efficiency, and other technology enablers which drives the long-term electricity usage.

The state of the art of models are methods used for short-term load forecasting are introduced in Chapter 2 as a literature review.

1.4 Knowledge Gap and Main Contributions

Although a diverse set of load forecasting techniques and methodologies has been studied, there are still knowledge gaps on the subject of secondary substations or low-voltage (LV) load forecasting [16] as follows:

- Load forecasting has been extensively applied either at the system/region level or at the building/point-of-delivery scope. Short-term load forecasting at the low-voltage level, other than at the smart meter level[1], such as secondary substations, has not been as extensive [16]. However, this does not mean that efforts have not been made. This research and references [17, 18] should be considered.
- Instead of evaluating the developed method only for performance (accuracy), practitioners should consider the impact and applicability of the method, as well, toward a live system implemented and adopted by the energy sector player.
- LV load forecasts can benefit from weather forecasts, particularly when these predictions are derived from multiple weather stations or from numerical weather predictions (NWP), whose significant advances have not yet been translated into improved LV load forecasts.
- In the context of LV load forecasting, numerous articles did not use benchmarks to compare their models beyond benchmark persistence forecast.
- Due to the scarcity of LV timeseries (Irish CER dataset and UK Low Carbon London trial data), one line of study would be to compare models using multiple datasets from different sources, rather than tackle only the few open datasets.
- There is a need for developing pragmatic methods to achieve the explainability of Artificial Intelligence (AI) and Machine Learning (ML) methods in the LV load forecasting

[1] Smart mert level means the low-voltage consumers as small buildings or houses.

context, as well as make the models computationally cheap in order to tackle thousands of low voltage assets and predict its load curve for the next few days in a useful time to business needs.

This study seeks to contribute with an approach specifically tailored to STLF at the secondary substation level, an area of research that has been sparse. With that goal in mind, the following steps are considered in depth detail: (i) evaluation criteria considering the applicability, interpretability, reproducibility and accuracy aspects, (ii) benchmark against a classical regression model for system-level forecasting described by Tao Hong and his research group [2], and (iii) modeling input variables associated with a consistent description of parameter tuning and alternatives from both the statistical and the energy domain point of view.

Using this strategy, it will be possible to apply STLF to a large number of different time series at secondary substation level — because the parameter tuning and method details chosen at the system level might change when applied to the others, a consistent description will keep the reproducibility —, keep a communication bridge between load forecasters, operators, planning manager, and executives when considering aspects more than accuracy, such as interpretability and applicability — note how the load forecaster developed needs to be approved by managers, understandable by operators, and defensible before the regulator —, and the analysis of the results when this systematic methodology is applied, for the first time, to the all 100,000 secondary substations which integrates the Portugal power grid. Figure 1.1 diagrams the scope of the study across the voltage levels of the power grid. In particular, the developed models are summarized in Table 1.6 and the methodology is detailed as follows:

- Using identical explanatory variables, we compare a Generalized Additive Model (GAM) with a classical one. Accuracy is assessed using different metrics while keeping in mind that a biased evaluation could result from choosing only one metric. This would serve as a benchmark base for LV load forecasting (Section 3.2);

- Develop a method to improve the GAM-based regression model by introducing new synthetic explanatory variables based on the same data. New variables are introduced based on domain knowledge and a systematic approach. Numerical weather predictions were used as explanatory variables in the low voltage context (Section 3.3);

- Compare the improved GAM-based regressor to a gradient boosting machine (GBM) as the XGboost implementation with the same explanatory variables after necessary adjustments and hyperparameter optimization but lacking interpretability (Section 3.4);

- Compare the improved GAM-based regressor to a simple new ensamble method, where weaker forecasters (also GAM-based) improve accuracy while still maintaining desirable interpretability (Section 3.5);

- Rather than tackle the few open datasets of secondary substations load, use of a new dataset that encompasses all 100,000 secondary substations of the Portuguese electricity grid, where energy is converted from medium voltage to low voltage using power transformers; train and evaluate those individual disaggregated load forecaters (Section 4.3);

- Develop an appropriate data representation of time series, such as the discretization into symbol sequences with the aim of keeping and highlighting daily shapes and regimes throughout the year, week, and public holidays (Section 6.4).

- Take advantage of clustering techniques to split this new dataset, projected as symbol sequences, into groups that contain load curves with similar daily shapes and patterns (Section 6.5).
- Compare individual disaggregated forecasters with the new cluster-based regression models which take advantage of clustering and daily discretization for the same dataset (Section 6.6);
- The methods and algorithms were implemented and adopted by Portugal's DSO as a live system capable of handling thousands of load curve predictions in a useful time and a facilitator of innovation (consider the impact of this research area); the architecture of the distributed IT system is described in Chapter 7;
- Describe the underdeveloped aspects in the context of LV load forecasting.

In addition, Chapter 5 is dedicated to exploring the classification of power load time series considering the daily shapes. Among several classification use cases, highlight the model which identifies the type of power consumption (household, industry, services, utilities, transportation) through the pattern of time series.

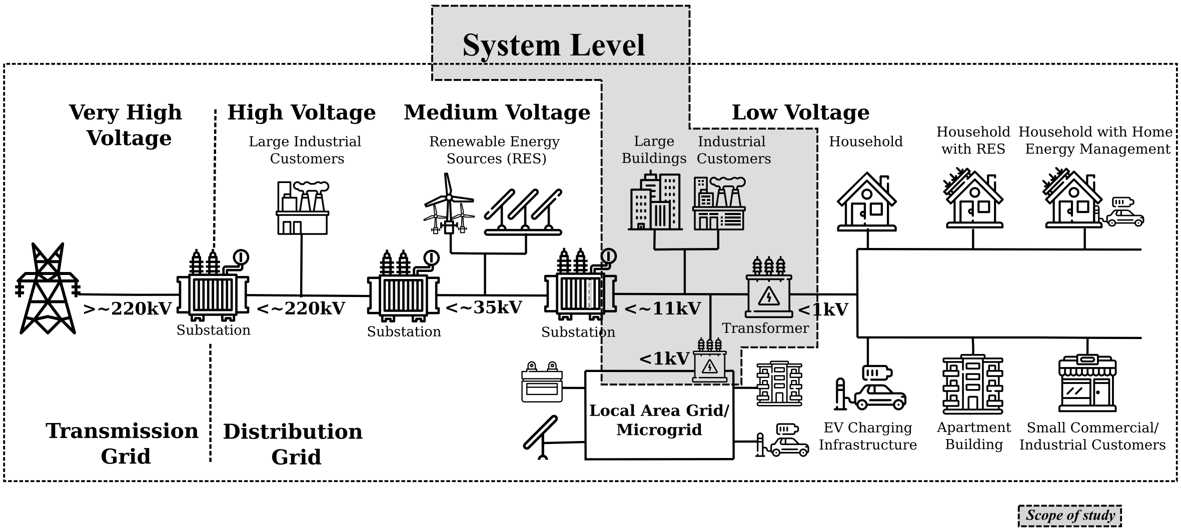


Figure 1.1. The scope of the study includes the system level and the secondary substations of Portugal DSO grid including the private secondary substations of large electricity consumers at high and medium voltage.

Part of this study has been published in two peer-reviewed journal papers. One paper focuses on the systematic approach to short-term load forecasting from the system level down to nearly 10,000 secondary substations [19], covering Chapters 2, 3, and 4. The other paper discusses the classification of power load timeseries using a technique called shapelets and presents the assessment of four different use cases [20], covering Chapter 5. Additionally, two other papers, which explore the practical applications of this study in industry, were published at the CIRED conference. These papers delve into the massive forecasting of timeseries in an enterprise system known as PREDIS [21, 22].

The reader might ask, given the huge efforts of the scientific community on the subject over several years, why their techniques have been mostly evaluated otherwise than against the secondary substation's time series. The point is that most Distributed System Operators (DSO)

Table 1.6. The study of several different models were conducted and organized in 4 chapters.

Section	Model	Goal	Dataset
3.2	GLMLF-B	System level GLM benchmark model	3.1
3.3.1	GAMLF-SL-M1	System level GAM model based on benchmark model	3.1
3.3.2	GAMLF-SL-M2	System level GAM model with lagged load covariates	3.1
3.3.3	GAMLF-SL-M3	System level GAM model with calendar covariates	3.1
3.4	GBMLF-SL	System level XGboost model	3.1
3.5	GAMLF-SLE	System level GAM and WMC-R ensemble model	3.1
4.3	GAMLF-SSL	Individual Secondary Substation Model	4.2
5.5.1	SHP-W	Shapelets Classification - Weekends	3.1
5.5.2	SHP-EMM	Shapelets Classification - Early Monday Morning	3.1
5.5.3	SHP-LDPS	Shapelets Classification - Load Dynamics in Substations	5.3
5.5.4	SHP-TPC	Shapelets Classification - Type of Power Consumption	4.2
6.4	CLULF-SSL-D	Cluster-based Forecasting - Discretization	4.2
6.5	CLULF-SSL-SSC	Cluster-based Forecasting - Symbol Sequences Clustering	4.2
6.6	CLULF-SSL	Cluster-based Forecasting	4.2

have just rolled out smart meters in secondary substations in the last three to six years [23–25] plus the time it takes to get a proper period of data. Furthermore, no or partial subsets of these data have been made public. In contrast, this study addresses the entire dataset that has been collected from secondary substations in Portugal's power grid since 2015.

Chapter 2

Literature Review

Thousands of papers, reviews, and reports on electricity forecasting have been written over the past 50 years. As a reference, there are nearly 7500 entries related to the topic of electricity load forecasting in the Web of Science Database[1] with a growing trend in the annual number of publications. Considering its magnitude, this chapter focuses on the most important literature based on either the reputation of the journal in which it was published or the number of citations and attention it has garnered.

Short-term load forecasting is a problem of time series forecasting applied to the energy domain. Surveys for time series forecasting have been published and usually grouped the models into (i) statistical regression models with classical equations, (ii) machine learning forecasting models, and (iii) hybrid forecasting models [15, 26, 27]. They also summarize the general structure of the models using mathematical equations, which will not be repeated here.

This chapter introduces representative reviews and surveys published in recent years, as well as state-of-the-art references to energy forecasting, following this structure: statistical regression-based models in Section 2.1, machine learning-based models in Section 2.2, and methodological approaches including hybrid models in Section 2.3, mostly for short-term though some papers might focus one various horizon periods. Section 2.4 defines the four aspects the resulting models would be evaluated, such as applicability, interpretability, reproducibility, and the 9 accuracy metrics.

A considerable number of comprehensive reviews have established the evidence in STLF for further research concerning the models and methods [28–38].

The first few papers about STLF were reviewed by Matthewman and Nicholson, in 1968, in which electricity demand was introduced as a time series with daily, weekly and yearly seasonal patterns, and exogenous variables were used to explain electricity demand, such as meteorological and calendar variables [28]. Two decades after, in 1987, Gross and Galiana covered the importance of STLF role in on-line scheduling and security functions of an energy management system [31]. They stated that the operating costs are reduced when the forecasting error decreases: once the load is less unpredictable, the reserve capacities of energy system may be reduced without affecting its reliability and security. They reviewed several techniques to predict the load shape and the peak. They found that pure time-of-

[1] http:\\webofknowledge.com

days models or similar day methods were being replaced by the dynamic models, which, by contrast, take on consideration the recent past of load, as well as, weather factors that influence load. Moghram and Rahman reviewed five widely applied techniques to STLF (multiple linear regression, stochastic time series, exponential smoothing, state space method, and knowledge-based approach) in terms of efficiency and difficulties of each one, rather than seek and enhance the best model [32]. Hahn, Meyer-Nieberg, and Pickl published a survey over a non-exhaustive set of 100 papers and reviews concerning load forecasting [34]. They have realised that there are various approaches applied to load forecasting ranging from regression-based methods over time-series approaches towards artificial neural networks and expert systems. In addition, they concluded that selecting the appropriate model depends on the problem and the situation currently under consideration and, therefore, “no general recommendations can be given”. Suganthi and Samuel made reference of more than 350 papers related to energy demand forecasting models with different goals, purposes and forecasting horizons organized by the technique applied, although did not compare the results of the diverse set of approaches [35]. Nti et al. reviewed 77 relevant papers from 2010 to 2020 with a concise summary of the useful characteristics of compared techniques as used method, timeframe, train and test split, error, accuracy metrics [29].

Nevertheless, as [36] points out, most studies pursue the goal of finding the best technique for load forecasting, resulting in rather worthless articles. Either because virtually most papers focus on showing the superiority of the introduced technique on very specific data sets hiding their weaknesses, or lack of detailed information on the setup experiments or over-manipulating the data (for instance, excluding days whose electricity load are more unpredictable) [30]. *A universally best technique simply does not exist*, it is the data and the business needs that determine which technique is more useful.

2.1 Statistical Regression-based Models

The classic forecasting method of time series is based on mathematical and statistical modeling. Some of the most widely used methods are the autoregressive techniques introduced by George Box and Gwilym Jenkins, such as the autoregressive moving average – **ARMA**. To use the ARMA model, an essential condition is that the time series should be stationary, which is achieved by differencing the non-stationary load time series in the first place. [39] is an example of applying ARMA to predict the next day system load, which contributed a multi-model partitioning filter (MMPF) to select the correct model order of ARMA in an STLF method which evolves a online adaptive procedure. Usually, the appropriate ARMA (and ARIMA) models are obtained by applying techniques to identify the order/parameters of the model, such as the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC).

A sequence that contains characteristics such as trend, seasonality, or periodicity is called a non-stationary sequence. Auto-regressive integrated moving average – **ARIMA** – is a well-known non-stationary time series model, which can reflect the changes of different data patterns, and the model requires fewer parameters to estimate. It is also necessary to determine the order before constructing the ARIMA model. Autocorrelation function (ACF) and partial autocorrelation function (PACF) are often used to determine the order of the ARIMA model. [40] is an example of applying ARIMA to system load forecasting. Additionally, it uses another non-statistical method to model the nonlinear component present in power load time series, in this case in the resulting residuals from the first ARIMA model.

To deal with seasonality, seasonal ARIMA – **SARIMA** was introduced. [41] is an example of applying SARIMA to predict the half-hour power load of the system one day in advance. It also contributes with robust models to detect half-hourly data points influenced by *non-normal days* and consider them as outliers.

Overall, these forecasting models are adaptable, can deal with seasonality and with non-stationary data, and only require the past value of a timeseries. However, it is unlikely to perform well on long-term predictions and they are unable to include exogenous variables that domain practitioners consider essential.

[42] used SARIMA to predict the hourly system load up to 4 weeks in advance, in which the temperature effects are captured through heating and cooling degree days. Better accuracy was achieved when temperature effects are captured through regression splines. In the same manner, the non-linear temperature effect is modeled using regression splines by [43]. The calendar, lagged temperature, and autoregressive electric load components are applied to a periodic ARMA model – **PARMA** – to make hourly forecasts of the electrical load from one to ten days ahead.

Statistical methods, such as generalized linear models – **GLM** – and generalized additive models – **GAM** –, have been used to model the relationship between variables, including exogenous explanatory variables. Thus, the outcome or dependent variable is defined by other variables called explanatory or independent variables. Different methods allow for diverse types of dependency/constraint modeling between the dependent and independent variables.

GLM is used to model the interactions of calendar and temperature as exogenous explanatory variables, and historical power load as auto-regressive explanatory variable to predict the hourly system load one day ahead [2, 44].

GAM is also used to model the interactions of historical power load, calendar, and additional meteorological variables [45, 46]. Splines, wavelets, and hybrid alternatives are compared as smooth functions for the additive components of GAM [47].

Probabilistic forecasting has also been applied to short-term load forecasting. Partially linear additive quantile regression – **PLAQR** – is used for short-term load forecasting in [48] combining it with the unit commitment problem during peak hours. Some researchers have been moving from the traditional deterministic decision-making framework to its probabilistic counterpart [36]. Incorporating the uncertainties of load forecasting as input to the goal analysis, such as load flow analysis, unit commitment problem, reliability planning, and energy price forecasting, has been recognized as a necessity for decision-making and risk taking. Kernel density estimation has also been used for probabilistic forecasting [49] and compared to the SVM and ANN versions [50].

Table 2.1 gives an overview of the references for statistical regression-based models considering the statistical technique(s) used, the object and forecasting horizon, the data sets in which the object and exogenous variables were extracted, and the evaluation aspects and metrics used to assess the resulting model. Although a diverse set of datasets and methods have been

Table 2.1. Overview of references for statistical regression-based models.

Ref	Models	Object & Horizon	Dataset	Evaluation
[39]	ARMA	Daily system load	2Y Hellenic Power	Accuracy: RMSE
[40]	ARIMA	Yearly system load	50Y China	Accuracy: MAE, RMSE, MAPE
[41]	SARIMA	Half-hourly system load, one-day ahead forecast	70W+30W[iv]	Accuracy: MAPE for *normal days* with outlier robustness, Applicability: fast execution and online implementation
[42]	SARIMA + MARS[v]	Hourly system load, up to 4 weeks ahead forecast	10Y South African load, including temperature data from 36 meteorological stations	Accuracy: MAE, RMSE, MAPE
[43]	PARMA + MARS	Hourly systme load, up to one to ten days ahead	12Y+1Y Spanish load, including temperature data	Accuracy: RMSE, MAPE
[44]	GLM	Hourly system load, one-day to one-week ahead forecast	3Y+1Y medium US Utility, including temperature data	Accuracy: MAPE (hourly, daily, daily peak, daily valley)
[2]	GLM	Hourly system load, one-day or one-week ahead forecast	3Y+1Y (different updating cycles: hour, day, week, year), including temperature data	Accuracy: MAPE (hourly, daily, daily peak, daily valley, ...), Applicability, Simplicity, Reproducibility
[45]	GAM	Hourly system load, one-day ahead forecast	5Y+1Y French, including temperature and cloud cover (1 month updating cycle)	Accuracy: RMSE for *normal days*
[46]	GAM + CLR[vi]	Half-hourly system load, one-day ahead forecast	13Y+1Y French, including temperature and cloud cover	Accuracy: RMSE, MAPE
[47]	GAM	Half-hourly demand load (one model per aggregation[vii])	1Y from CER dataset, including meteorological data and client classification	Accuracy: RMSE
[48]	PLAQR[viii]	Hourly system load focused on peak hours,	3Y+6M South African including calendar and meteorological data	Accuracy: MAE, RMSE, MAPE, CRPS[ix], LogS[x], Pinball loss[xi]
[49]	KDE[xii]	Hourly LV consumption	9M+3M active power from 103 LV spanish consumers	Accuracy: MASE, CRPS, Applicability

[iv] The two numbers and symbols means respectively the size of training and testing dataset; "Y" for years and "W" for weeks. [v] Multivariate Adaptive Regression Splines [vi] Curve Linear Regression [vii] Aggregation size is parameterized, 10 to 500 consumers for a total of 4623 consumers (residential customers and small-to-medium enterprises) in the dataset. [viii] Partially linear additive quantile regression combining GAM and quantile regression (QR) [ix] Continuous rank probability score [x] Logarithmic score [xi] also known as quantile loss function [xii] Kernel density estimation

applied to STLF, the object of the forecast is mainly the national or regional electric load and not the secondary substations load forecasting.

2.2 Machine Learning-based Models

Machine learning-based models have also been applied in STLF, such as artificial neural networks (ANN) [2, 51–53], long short-term memory based (LSTM) [54, 55], temporal fusion transformers-based (TFT) [56], support vector machines (SVM) [57, 58], fuzzy logic [59–61], random forest [62], particle swarm optimization (PSO) [63, 64], genetic algorithms (GA) [55], and others that might combine more than one technique.

Bogomolov et al. used general public dynamics derived data from cellular network and the energy consumption dataset to predict the next 7 days of energy demand from a northern region of Italy. The **random forest** regression technique was applied to predict the local electricity demand. Beyond the exogenous calendar and weather features that other studies pore over, this one demonstrates the most important features of more than 3000 synthetic variables derived from aggregated incoming and outgoing calls, received and sent SMS, and Internet connection events generated every 10 minutes within each square of the partitioning grid [62].

Regarding **artificial neural network**-based forecasters (ANN), most articles used the multi-layer perceptron, usually a feed-forwarded fully connected, or alternatively used a recurrent network. The activation function (either logistic or hyperbolic tangent functions) and the number of hidden layers (mostly one or two) are the two most common points. On the contrary, there are differences in choosing the number of input, output, and hidden neurons. The number of output neurons depends on the methodology used to forecast the 24-hour load profile. In iterative forecasting, one ANN is used to forecast one hourly load at a time, so that the forecasts for later hours will be based on the forecasts for the earlier ones in a multi-step fashion. In multi-model forecasting, 24 different models, one for each hour of the day, are used in parallel to forecast the 24-hour profile. This method is also common for load forecasting with regression models and has the advantage that the individual ANN are relatively small and not likely to be overfitted, one of the problems to be aware when using ANN for load forecasting. In single-model multivariate forecasting, all the load 24 hour profile is forecasted at once, resulting in an ANN with 24 output neurons or more if a half-hour profile is needed. Although this method was used by most of the articles reviewed, MLPs must be very large to accommodate 24 output neurons, and depending on the number of input neurons, the number of parameters will be very likely to run into the thousands. On the other hand, treating each day as a vector means that one year of that will yield only 365 data points, which seems to be too few for the large MLPs required. The selection of input neurons is rather dependent on the *a priori* knowledge of the behavior of the system under study and the factors that influence the load. There is little theoretical basis for that decision. The same occurs when selecting the number of hidden neurons. In most articles, the authors have chosen this number by trial and error for better accuracy. The number of hidden neurons must be balanced to be flexible and powerful enough to fit the data and not too overly emphasized that it will overfit. Cross-validation or regularization techniques would avoid overfitting [65]. Hong evaluated several ANN and GLM forecasting models in which a new set of input features was added throughout the modeling approach. Using the same dataset, the models were compared by their precision. The number of hidden neurons was optimized as a hyperparameter through a range search [2].

Kong et al. propose a framework based on **long short-term memory** (LSTM) recurrent neural networks for residential load forecasting [66]. The proposed framework was tested on a 3-month half-hourly energy consumption of 69 residential consumers. As a result, the proposed LSTM approach performs better compared to various alternative state-of-the-art approaches. It turns out that many load forecasting approaches which are successful for system or substation load forecasting struggle in the single-meter load forecasting problem where high inconsistencies in daily consumption profiles generally affected the predictability. Kong et al. concluded that the higher the inconsistency, the more LSTM can improve forecasting compared to simple backpropagation neural networks. Furthermore, although individual load forecasting is far from accurate, aggregating all individual forecasts yields a better forecast for the aggregation level compared to the conventional strategy of directly forecasting the aggregated load.

Fuzzy regression considers that the deviations between the observed values and the estimated values are assumed to be dependent on the indefiniteness of the system structure, while in multiple linear regression the deviations are supposed to be errors in the observed values. There are examples that improve the precision of equivalent multiple linear regression [60].

Genetic Programming (GP) exploits the concept of evolution to tackle the search for possible model structures (or any computer program) and perform symbolic regression. Can be employed to search complex linear spaces. When selecting input variables, GP automatically finds the variables that contribute the most to the model and then constructs an equation [40].

Raza and Khosravi reviewed the characteristics, explanatory variables and importance of load forecasting, but mainly, they discussed the application of artificial neural networks as a superior performance over statistical techniques mainly when abrupt changes in environmental or sociological variables occur [67]. One interesting point of this reference, they explored the hybrid techniques that were proposed by published research with the combination of superior attributed of two or more algorithms. They concluded that the hybridisation of two or more techniques shows better results for load forecast problem than one technique alone, either conventional statistical techniques or ANN. They refer to: (i) ANN with fuzzy and genetic algorithm, (ii) ANN with expert system and regression techniques, (iii) ANN with wavelet and time series, (iv) ANN with support vector machine and artificial immune system, (v) ANN with genetic algorithms, (vi) ANN with gradient based learning techniques. Besides those hybrid optimisation techniques to be investigated in future directions, the authors pointed out meteorological factors to be considered besides temperature, as well as, electricity price as an important influential parameter on load demand in deregulated electricity markets , and solar distributed generation and demand side energy management that influence the net energy demand from grid.

Machine learning-based techniques have been developed and applied in diverse applications, and some highlighted their superior capability to handle complex input and output relationship, mainly the non-linear correlations. Table 2.2 gives an overview of the references for machine learning-based and hybrid models.

Table 2.2. Overview of the references for machine learning-based models and hybrid models.

Ref	Models	Object & Horizon	Dataset	Evaluation
[2]	ANN	Hourly system load, one-day ahead forecast	3Y+1Y (rolling 9Y of data)	Accuracy: MAPE, Interpretability compared to GLM alternative
[53]	ANN, Bagging	Hourly system load, one-day ahead forecast	4Y+2Y New England Pool data, including temperature and calendar	Accuracy: MAPE, Applicability: Computational time
[54]	LSTM, ResNet	15min demand load (one model to many), one-day ahead forecast	473D+60D, 36 Korean HV consumers	Accuracy: MAPE
[56]	TFT	Hourly consumption, one-day ahead forecast	7M+1M UCI dataset (369 consumers)	Accuracy: P50 and P90 quantile loss, Interpretability: variable importance, persistent temporal patterns, regimes and significant events
[58]	SVM	Hourly System Load, one-day ahead forecast	27M+2M, Inner Mongolia Power Grid dataset, including calendar and weather	Accuracy: Relative Error, RMSRE
[59]	Fuzzy Logic	Hourly system load, two-day ahead forecast	3Y, including temperature	Accuracy: RMSE, R^2
[60]	Fuzzy Regression	Hourly system load, one-day ahead forecast	2Y+1Y ISO New England (1 day updating cycle), including temperature	Accuracy: MAPE (hourly, daily, daily peak)
[62]	Random Forest	Average daily and peak daily local consumption up to 7 next days	2M Italy region consumption and people dynamics derived from square partitioning cellular network data	Accuracy: MAE, MSE, RMSE, RSE, RAE, R^2
[68]	Hybrid	Hourly regional level and Quarter-hourly small area level, one-day ahead forecast	2Y Tianjin region	Accuracy: MAE, RMSE, MAPE, Stability: Variance and Direction Accuracy (DA)

2.3 Methodological Approaches and Hybrid Models

Apart from focusing on the best technique, many articles also demonstrate how a methodology is used to solve the load forecasting problem or its subproblems. The general methodological approaches were identified into four general categories [36] and comprehensively reviewed after with the fundamental benefits and drawbacks [38]. Table 2.3 outlines the six general categories and techniques.

Table 2.3. Short-term load forecasting methods based on [36, 38, 68].

Method	Description
Similar Pattern • Similar Day • Pattern Sequence • Sequence Learning	Determines the load curve as a sequence of various similar load profiles.
Variable Selection • Stepwise Method • Correlation • Mutual Information • Filtering • Optimization Algorithm • Time-dependent	Presumes the load curve behaves like a series of variables either correlated or independent from each other.
Hierarchical Forecasting • Bottom-up • Top-down • Ensemble • Weight Combination	Considers the data as an aggregated load, which is highly varying by changes in the load at lower levels of hierarchy.
Weather Station Selection • Average Model • Optimal-number-of-stations Model	Determines the best-fitted weather data into the load model.
Decomposition • Wavelet Transform • Empirical/Variational/Dynamic Mode Decomposition • Singular Spectrum Analysis • Double-layer Decomposition	Decompose and extract the characteristics of load time series before modeling them
Error Correction • Error forecasting	Error correction techniques extract useful information from the error values to correct the predicted values

Classifying the consecutive daily load prior to time series forecasting results in a reduction in forecast error and eliminates the need to explicitly decompose the curve prior to the regression task [46].

Kong et al. developed a forecasting method based on error correction using dynamic mode decomposition (DMD) for STLF, including data selection, error forecasting, and error correction. In the data selection stage, three types of data are selected as input data of the model, including previous day data, same day data in previous week, and similar day data obtained by grey relational analysis (GRA). In the error forecasting stage, the data driving characteristics of the DMD algorithm are used to capture the potential spatiotemporal dynamics of error series, thereby realizing the error forecasting. In the error correction stage, on the basis of combining the forecasting results of load and error, an extreme value constraint method (EVCM) is developed to further correct the load demand series. The article provides a stable and accurate error correction method for the load forecasting model and demonstrated improvement for a diverse set of forecasting techniques [68].

In order for a system to handle a huge volume of time series forecasting, various methods have been followed, such as stream-based load forecasting [69] and big data analytics [70]. There are open-source data science toolkits for energy, such as GridDS and Linux Foundation for Energy projects. By providing an integrative software platform to train and validate machine learning models, GridDS will help improve the efficiency of distributed energy resources, such as smart meters, batteries, and solar photovoltaic units. Linux Foundation for Energy sets the foundations for open source collaboration, which includes the OpenSTEF project for energy forecasting.

2.4 Evaluation Criteria

In addition to the accuracy criteria to which most articles refer, the models are evaluated from three other aspects. These are the four aspects:

- **Applicability** – The model uses data and information that the utility is able to obtain with tangible resources (both human, data, and system resources). For example, if a utility is unable to get an up-to-date calendar of special events, such as strikes, conferences, sports, and cultural events, which have an impact on the power load at the secondary substation that feeds the infrastructure or venue, then a model containing that variable is not applicable, no matter how well that variable would improve the predictions.

- **Interpretability** – The degree to which a human can understand (mental model) and consistently deduces the result of the model [71]. When the model is more interpretable, humans can more easily understand why certain outcomes are realized and consequently accept them easily. Note that the term *explainability* is used rather to refer to the explanations of individual predictions.

- **Reproducibility** – The model and the method carried out to build the model are systematic and documented, and it is possible to replicate them in other time horizons, geographies, or grid levels. If engineering heuristics are involved, the tweak and tuning process should be well defined.

- **Accuracy** – The performance of the model in terms of how close the predictions are to the real values. In particular, the mean absolute error (MAE), the mean absolute percentage error (MAPE), the root mean square error (RMSE), the normalized RMSE (NRMSE), the coefficient of determination R^2, and the mean absolute scaled error (MASE) [72]. Note that these accuracy measures focus on smoothness of the forecasts, rewarding them. At the distribution level, it is the peak that matters for many use cases. So, for secondary substation models, we used Haben's adjusted error and a normalized version that prefer models that predict peaks even within a restricted displacement that do not forecast at all [73]. Additionaly, the timeseries cross-validation procedure is used to assess performance, that is, the corresponding training set consists only of observations that occurred before the observation that forms the test set [74].

Let y_t be the real value and $\hat{y}_t$ the prediction at time t, define the error metrics for the period $[1, T]$ as

$$e_{\text{MAE}} = \frac{1}{T}\sum_t \left|y_t - \hat{y}_t\right|, \tag{2.1}$$

$$e_{\text{MAPE}} = \frac{1}{T}\sum_t \left|\frac{y_t - \hat{y}_t}{y_t}\right|, \tag{2.2}$$

$$e_{\text{RMSE}} = \sqrt{\frac{1}{T}\sum_t \left(y_t - \hat{y}_t\right)^2}, \tag{2.3}$$

$$e_{\text{NRMSE}} = \frac{\sqrt{\frac{1}{T}\sum_t \left(y_t - \hat{y}_t\right)^2}}{\frac{1}{T}\sum_t y_t}, \tag{2.4}$$

$$e_{\text{R}^2} = \frac{\sum_t \left(\hat{y}_t - \bar{y}\right)^2}{\sum_t \left(y_t - \bar{y}\right)^2}, \tag{2.5}$$

$$e_{\text{MASE}} = \frac{\sum_t \left|y_t - \hat{y}_t\right|}{\sum_t \left|y_t - y_{t-m}\right|}, \tag{2.6}$$

where m is the seasonal period for executing the naive m-step seasonal forecast method, $\hat{y}_t^{\text{NAIVE}} = y_{t-m}$. If the timeseries presents more than one seasonality, it is compared to the seasonality with the lowest period. This latter error metric is scaled-free and is well suited, unlike the MAPE, to timeseries with zero or near zero values because it never gives infinite or undefined values, except in the irrelevant case where $\forall_t \; y_t = C$.

Haben's adjusted p-norm error (APN), the mean adjusted p-norm error (MAPN), and the normalized mean adjusted p-norm error (NMAPN), are defined as

$$e_{\text{APN}} = \min_{\mathbf{P}\in\mathscr{P}^{(w)}} \|\mathbf{P}\hat{\mathbf{y}} - \mathbf{y}\|_p \;, \tag{2.7}$$

$$e_{\text{MAPN}} = \sqrt[p]{\frac{1}{T} {e_{\text{APN}}}^p} \;, \tag{2.8}$$

$$e_{\text{NMAPN}} = \frac{\sqrt[p]{\frac{1}{T} {e_{\text{APN}}}^{p}}}{\frac{1}{T} \sum_t y_t} \,, \tag{2.9}$$

where $\mathscr{P}^{(w)}$ is the complete set of restricted permutations such that $P_{ij} = 0$ for $|i - j| > w$, restricting the magnitude of the displacements of the forecast values. Haben suggests using the absolute 4-norm error, $p = 4$, to penalize large errors (i.e., missed peaks) much more than small errors. The choice of the adjustment limit w depends on the use case. Nevertheless, for general purposes, this study uses $w = 3$, which means that forecasts can be displaced by up to 3 half hours on either side of their original forecast time.

2.5 Conclusions

The literature review demonstrates the evolution of the STLF subject with the application of many statistical regression-based and ML-based techniques and other hybrid techniques to solve the STLF challenge and its subproblems (similar pattern, variable selection, hierarchical forecasting, selection of weather stations, time series decomposition, and forecast error correction).

Additionally, the growth of data and computing power available make possible the use of more explanatory variables, as well as more complex techniques to preprocessing data, modeling the data, optimizing the hyperparameters or, even, optimizing or searching over the solution space.

However, as the literature has been reviewed, it became increasingly evident that the knowledge gaps described in Section 1.4 have not been adequately addressed. The articles and surveys pointed to are mostly for the forecasting of system load, and in fact STLF at the low voltage level, other than at the smart meter level, such as secondary substations, has not been as extensively studied. In addition, most studies pursue the goal of finding the best technique for load forecasting by comparing the accuracy, with less focus on interpretability, applicability, and reproducibility, including setup. Even, some studies have challenges due to the small dataset for testing purposes (less than one complete year for a problem with yearly seasonality), or lack of detailed information on the setup experiments, which diminish their reproducibility.

Chapter 3

National Load Forecasting

The national power load denotes all electricity demand by consumers and energy storage systems, plus the part that is naturally lost during energy transmission and distribution. In addition, it can be seen as the amount of energy produced in centralized power stations (hydro, solar, wind, natural gas, coal, oil, nuclear, and others) added to the net electricity imported from neighboring countries.

This Chapter covers a set of contributions in the short term load forecasting at the national level. The subject of analysis is the total load of the system minus the power used in pumped hydroelectric storage systems, and the data set is overviewed in Section 3.1. The benchmark model is established and is based on the classical regression model purposed by Tao Hong and his research group (Section 3.2). This is compared first with a GAM-based regression model using identical explanatory variables and second with enhanced models using the same technique (GAM) but introducing new synthetic explanatory variables extracted from the same data (Section 3.3). Finally, gradient boosting (Section 3.4) and a purposed ensemble method (Section 3.5) are compared.

3.1 Data Overview

In order to forecast the national power load, one follows a machine learning approach with a real dataset publicly available and downloadable from the TSO Data Hub website[1].

Let consider the dependent variable **y** as the national power load,

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_t & \dots & y_T \end{bmatrix}^\top . \tag{3.1}$$

Each entry y_t denotes the national (mainland Portugal) power load in megawatt (MW) at time t which comprises 30 minutes period. The resolution of original data is quarter-hourly, but for modeling purposes, every two sequential data points were aggregated using the average to downsample electrical power time series.

The power load follows a pattern with annual, weekly, and daily seasonality. Figure 3.1 shows the half-hour demand for a week from Monday to Sunday, in winter and summer. Note the weekly pattern: Working days are very similar, while Saturday and Sunday are different in both level and form. On working days, the three peaks are different depending on the season.

[1] REN Data Hub website: https://datahub.ren.pt/

Moreover, there are some important decreases in demand during the holidays of Christmas, the summer holidays, and the bank holidays.

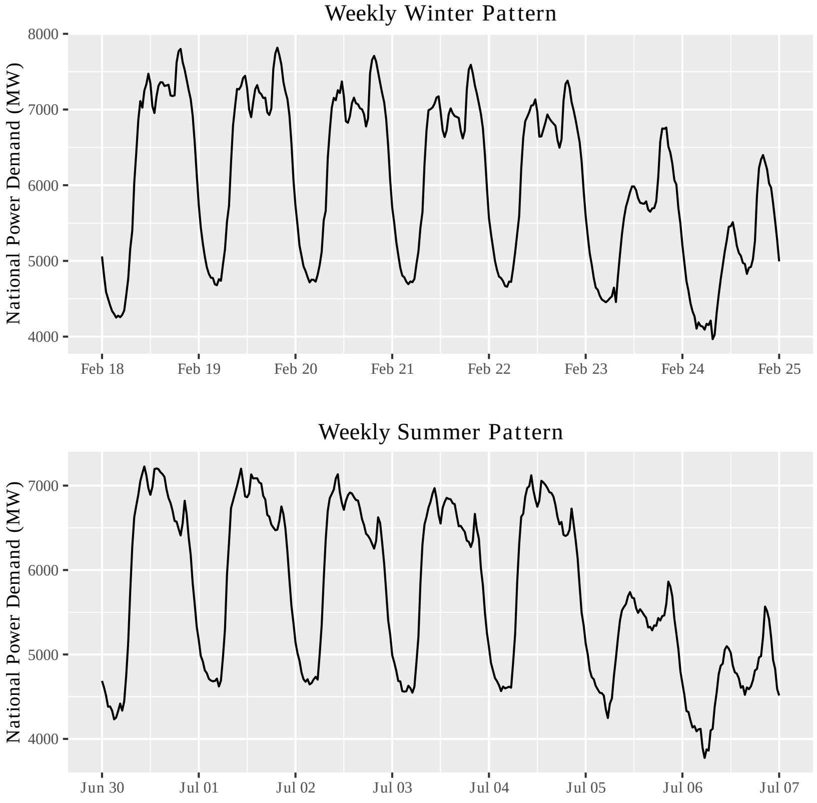


Figure 3.1. Portuguese demand from Monday to Sunday, in February 2008 and July 2008.

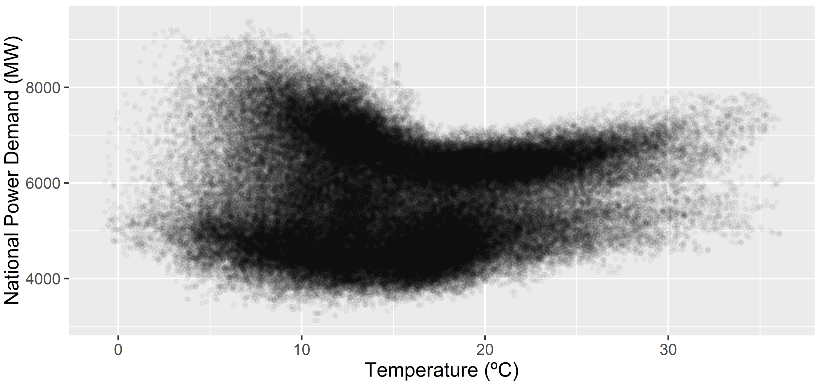


Figure 3.2. Half-hour national demand scattered plot with temperature observations.

Additionally, weather observations from 12 selected weather stations in Portugal are included in this dataset to study the effect of temperature on power load. Figure 3.2 exhibits the nonlinear dependence of the power load on the temperature due to electrical heating and cooling.

Weather data are publicly available and can be downloaded from the NOAA website[2]. The time resolution is different from station to station: While some record the temperature

[2] NOAA Data Access website: http://www.ncdc.noaa.gov/data-access

every 3 hours, other stations record every half-hour. The dataset includes meteorological observations such as temperature, dew point temperature, sea-level pressure, observable weather on a qualitative scale, and more. Data wrangling was performed to mitigate missing values, cope with temporal resolution differences between weather stations, and summarize the 12 weather station time series into one. Accordingly, the following steps are taken.

As a first step, the temperature values are removed when the temperature quality field, a metadata presented along with the temperature field, indicates that specific values might be incorrect or suspicious for a given timestamp.

Second, each weather station's raw observations are grouped into half-hour intervals. It should be noted that weather stations can simultaneously generate multiple reports of observations for different purposes. Additionally, the reporting method has evolved over the period of data collection (some reports have ceased to be issued or its periodicity has changed). To simplify these details, the weather fields are derived from averaging the multi-value within each half-hour period.

As a third step, a weighted average of the temperatures of the 12 weather stations is used to calculate the "national temperature". The weights reflect the 2013 annual electricity consumption of each region. This information was obtained from PORDATA's 2013 electricity consumption data for Portugal[3]. Table 3.1 shows the 12 selected weather stations and the weights applied to each region.

Finally, linear interpolation between two non-missing values reduces time series gaps. Due to temperature dynamics (linear behavior over a short period of time), that linear interpolation is only applied when two non-missing values did not distance more than 2.5 hours. If a three- or more-hour gap occurs, the data are not interpolated there.

Table 3.1. The weighted average of "national temperature" is computed from measurements of 12 selected weather stations whose weights are according to the power demand of each region. In regions with more than one weather station selected, a simple average is calculated between them.

Region	Weight	Weather stations
Algarve	6.8%	Faro
Alentejo	7.8%	Beja and Portalegre
Lisboa	26.8%	Lisboa and Lisboa/Gago Coutinho
Centro	25.3%	Coimbra, Viseu, Castelo Branco, and Monte Real
Porto	15.2%	Porto
Norte	18.1%	Vila Real and Bragança

[3] http://www.pordata.pt

3.2 Benckmark Model – GLMLF-B

The generalized linear model-based load forecasting – benchmark model, GLMLF-B, is a classical regression model for system-level forecasting described by Tao Hong and his research group[2]. The model uses generalized linear regression, which can be defined as

$$g(\mu_t) = \beta_0 + \beta_1 x_t^{(1)} + \beta_2 x_t^{(2)} + \dots \tag{3.2}$$

where $\mu_t \equiv \mathbb{E}(Y_t)$, Y_t is a response variable and $Y_t \sim$ some exponential family distribution, g is the link function which provides the relationship between the linear predictor (right side of equation) and the mean of the distribution function, β_i's are unknown parameters or coefficients, and the variables x_j can come from different sources (quantitative inputs; transformations of quantitative inputs, such as log, square, and square-root; numeric or "dummy" coding of the levels of qualitative inputs; or interactions between variables, for example $x_t^{(3)} = x_t^{(1)} \cdot x_t^{(2)}$).

3.2.1 Model

In addition to the national power load y_t (Equation 3.1), available data $\mathbf{x}_t$, which can be used to explain the dependent variable, include trend index, calendar, time, and meteorological data as follows:

$$\mathbf{x}_t = \begin{bmatrix} x_t^{(\text{Trend})} & x_t^{(\text{Month})} & x_t^{(\text{DayOfWeek})} & x_t^{(\text{TimeOfDay})} & x_t^{(\text{Temperature})} \end{bmatrix}^\top . \tag{3.3}$$

Let explain these components in more detail:

- $x_t^{(\text{Trend})}$ is a quantitative variable that represents the index for the entire range of available data. For example, 1 for the first half-hour of available data, 2 for the second half-hour, etc. This variable is useful for capturing the trend of increasing or decreasing power loads. The use of this variable might pose problems in model applicability due to the long trend that might change in significant events, for example, merging two utilities, recessions, economic booms, pandemics, and so on.
- $x_t^{(\text{Month})}$ is the index of the current month within the year from 1 to 12.
- $x_t^{(\text{DayOfWeek})}$ is a categorical variable representing the day of week – one category for each day: 1 for Sunday, 2 for Monday, 3 for Tuesday, 4 for Wednesday, 5 for Thursday, 6 for Friday, and 7 for Saturday. Some articles combine Tuesday, Wednesday, and Thursday into the same category, but they do not scientifically explain why. Nevertheless, the empirical reason may be the fact that days without a weekend immediately before or after have a similar load pattern. Note that different cultures have different rest day schemes.
- $x_t^{(\text{TimeOfDay})}$ is the index of the current time of day. Since this index represents each half-hour, its values range from 1 to 48 representing respectively the midnight and 11:30 p.m.
- $x_t^{(\text{Temperature})}$ is the meteorological covariate that represents the half-hourly "national temperature" – a weighted average computed from 12 weather stations in Portugal as explained in Section 3.1.

Therefore, the benchmark model can be written as follows:

$$
\begin{aligned}
\hat{y}_t = {} & \beta_0 + \beta_1 x_t^{\text{(Trend)}} + \\
& + \sum_{i \in \{\text{January},\ldots,\text{December}\}} \mathbf{1}_{\left(x_t^{\text{(Month)}}=i\right)} \beta_{2i} \\
& + \sum_{j \in \{\text{Monday},\ldots,\text{Sunday}\} \times \{0,\ldots,47\}} \mathbf{1}_{\left(x_t^{\text{(DayOfWeek}\times\text{TimeOfDay)}}=j\right)} \beta_{3j} \\
& + \sum_{k \in \{\text{January},\ldots,\text{December}\}} \mathbf{1}_{\left(x_t^{\text{Month}}=k\right)} \left(\beta_{4k} x_t^{\text{(Temperature)}} + \beta_{5k} x_t^{2\,\text{(Temperature)}} + \beta_{6k} x_t^{3\,\text{(Temperature)}}\right) \\
& + \sum_{m \in [0,47]} \mathbf{1}_{\left(x_t^{\text{TimeOfDay}}=m\right)} \left(\beta_{7m} x_t^{\text{(Temperature)}} + \beta_{8m} x_t^{2\,\text{(Temperature)}} + \beta_{9m} x_t^{3\,\text{(Temperature)}}\right) + \epsilon_t
\end{aligned}
\quad . \tag{3.4}
$$

Let explain the model structure in more detail:

- The intercept β_0 models the base power load;
- $\beta_1 x_t^{\text{(Trend)}}$ take into account the long-term linear trend in the power load;
- β_{2i} and β_{3j} take into account the seasonal blocks monthly, weekly, and intraday in the load series. Each block is treated individually and non-continuously in this model structure, that is, they are qualitative variables;
- The last two additive components model the effect of temperature on the load as exhibited in Figure 3.2 using the 3rd order polynomials of the temperature whose parameters β are independently computed for each month and for each time of day.

3.2.2 Results

The results for GLMLF-B are shown in Table 3.2 for different update cycles. The metrics are computed through the following procedure: (i) over the time span from 2016 and 2019, take the three consecutive years as training data and the next period of updating cycle (one day, one week, two weeks, and one year) as testing data; (ii) roll the actual data of this testing period to the training data and recalculate the model; (iii) recompute the metrics with this new model, and so on, until all the periods in the year 2019 are forecasted. Table 3.7 has the detailed metrics including minimum, maximum, second, and third quantiles, as well as median and mean for time series cross-validation folds.

Table 3.2. GLMLF-B results calculated over the time span from 2016 and 2019 using times series cross-validation with a fixed 3-year window for training and the next 1 year, 2 weeks, 1 week or 1 day of testing data. The best results are in bold. The MASE is calculated with m steps equal to 52 weeks, 2 weeks, 1 week, and 1 day, respectively. All metrics are the mean calculated from cross-validation folds.

Update Cycle	**Folds**	**MAE** (MW)	**MAPE** (%)	**RMSE** (MW)	**NRMSE** (%)	**R^2**	**MASE**
1 year	1	235.25	4.23	353.12	6.14	0.888	0.519
2 weeks	26	217.75	3.87	296.99	5.17	0.918	0.742
1 week	52	214.64	3.82	283.73	4.96	0.929	0.970
1 day	365	**210.46**	**3.75**	**249.74**	**4.44**	**0.958**	1.207

The GLMLF-B model uses trend index, calendar, time, and temperature variables. All variables are acceptable by domain experts because of their interpretability: a quantitative variable (trend) to capture the increasing or decreasing trend of the power load over the interval, calendar variables to capture different effects in months and days of the week, time to capture intraday seasonality, and the effect of temperature on the power load. The result of the GLM fitting returns the weights of each parameter β associated with each explanatory variable, as defined by Equation 3.4. The model is highly interpretable because the weights manifest the degree of importance or effect of each variable. For example, one of the weights shows the effect of i^{th} hour on the power load and how that compares to the same hour on the previous day of the week.

From the applicability perspective, the use of a trend index obligates the refitting of the model as periodically as the change of that trend. Otherwise, the performance of the model rapidly degrades. Moreover, the use of temperature values observed at the exact time step for which the power load is forecast forces the availability of the temperature forecast as input to the model. Thus, temperature prediction errors could potentially decrease the performance of the power load predictions traced in this study.

3.3 Enhanced Model – GAMLF-SL

The generalized additive model-based load forecasting, GAMLF-SL, is a regression model for the system level. Generalized additive models (GAM) [75, 76] involve a sum of smooth functions of covariates, thus capturing the non-linear effects of covariates on the dependent variable. They have the form

$$g(\mu_t) = \beta_0 + f_1(x_t^{(1)}) + f_2(x_t^{(2)}) + f_3(x_t^{(3)}, x_t^{(4)}) + \ldots \tag{3.5}$$

where $\mu_t \equiv \mathbb{E}(Y_t)$, g is a smooth monotonic link function, $Y_t \sim$ some exponential family distribution, Y_t is a response variable, and f_j are smooth functions of covariates $x_t^{(k)}$.

The model offers a flexible specification of the dependence on covariates by specifying the model only in terms of “smooth functions”. It is able to capture complex non-linear relationships, and their estimation and prediction are straightforward. Additionally, GAMs have an important feature for contexts where domain experts need interpretable models: The simplicity of its additive structure makes it easy to use and understand.

There are several methods to estimate GAM, one of the most famous is the backfitting algorithm from Hastie and Tibshirani [76], which is implemented in the *gam* R package. Another method to estimate GAM is the Penalized Iterative Re-Weighted Least Square (P-IRLS) from Wood [77], which is implemented in the *mgcv* R package. In this method, the basis for each smooth function f_j is specified using regression splines of one or more variables. Given such a basis, a GAM can be estimated as a GLM, and, to avoid overfitting, the method controls the smoothness for each term through a set of penalties applied to the likelihood of the GLM.

3.3.1 Model

In addition to the dependent variable $\mathbf{y}$ already defined by Equation 3.1, consider these redefined data $\mathbf{x}_t$ that include calendar, time, and meteorological components as:

$$\mathbf{x}_t = \begin{bmatrix} x_t^{(\text{Trend})} & x_t^{(\text{DayOfWeek})} & x_t^{(\text{PublicHoliday})} & x_t^{(\text{DayOfYear})} & x_t^{(\text{TimeOfDay})} & x_t^{(\text{Temperature})} \end{bmatrix}^\top . \quad (3.6)$$

Note that these data exclude the month variables used in the previous section. Besides the components explained above, let detail the new ones:

- $x_t^{(\text{PublicHoliday})}$ is a categorical variable representing the national holidays, including Carnival; a category for each public holiday. Some papers combine all public holidays into a unique boolean variable. This option may be more prudent due to the minority number of public holidays over one year. We have evaluated the two options, but the final model takes into account the most descriptive categorical variable. Some public holidays have temporarily ceased from the Portugal calendar and that information was included. Regional holidays have an influence on national load patterns, but were not included.

- $x_t^{(\text{DayOfYear})}$ is a numerical variable representing the current day within the year. Its values range from 0 for 1st January and 1 for 31th December.

Considering the data available, the modeling activity requires to define what is the best combination of available input variables, following the additive structure which GAM follows. Therefore, the methodology carried forward has four steps: (i) formulation and selection of input variables, (ii) definition of model structure, (iii) model calibration and tuning, and (iv) evaluation of model and residuals. After residual analysis, the process is repeated to find a new set of input variables and model structure, focusing on the aspects or moments in which the residuals are higher. The process ends when a good balance is achieved between applicability, interpretability, reproducibility, and accuracy performance.

First, the definition and selection of the input variables was explained above, when the available data $\mathbf{x}_t$ were defined in Equations 3.3 and 3.6. Even though, Sections 3.3.2 and 3.3.3 detail the domain knowledge that reflects the definition and selection of additional input variables.

Second, the initial structure of the model, M1, was based on the GLMLF-B (Equation 3.4 with the necessary modifications due to the new characteristics of the GAM technique. Namely, the qualitative variables *month* and *time of day* are changed to their quantitative versions, that is, the quantitative *day of year* and *time of day*[4] variables, and smooth functions are used when appropriate. For instance, it is unnecessary to make explicit third-order polynomials for the temperature effect, since the smooth functions take care of this. Therefore, the GAMLF-SL-M1 model is defined as

[4] Here, the name (*time of day*) refers to the quantitative version. The interpretation is implicit from the context.

$$\begin{aligned}
\hat{y}_t^{(\text{M1})} &= \beta_0 + \beta_1 x_t^{(\text{Trend})} \\
&\quad + f^{(\text{TimeOfDay})}\left(x_t^{(\text{TimeOfDay})}\right) \\
&\quad + \sum_{i\in\{\text{Monday},\ldots,\text{Sunday}\}} \mathbf{1}_{\left(x_t^{\text{DayOfWeek}}=i\right)} \left(\beta_i + f_i^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)\right) \\
&\quad + f^{(\text{DayOfYear})}\left(x_t^{(\text{DayOfYear})}\right) \\
&\quad + f^{(\text{Temperature})}\left(x_t^{(\text{Temperature})}\right) \\
&\quad + f^{(\text{Temperature/TimeOfDay})}\left(x_t^{(\text{Temperature})}, x_t^{(\text{TimeOfDay})}\right) \\
&\quad + f^{(\text{Temperature/DayOfYear})}\left(x_t^{(\text{Temperature})}, x_t^{(\text{DayOfYear})}\right) \\
&\quad + \epsilon_t
\end{aligned} \quad . \tag{3.7}$$

Note that $f_i^{(\text{TimeOfDay/DayOfWeek})}$, $f^{(\text{Temperature/TimeOfDay})}$, and $f^{(\text{Temperature/DayOfYear})}$ introduce the co-interactions appropriately, excluding their main effects. For instance, $\forall_{i\in\{\text{Monday},\ldots,\text{Sunday}\}}$ $f_i^{(\text{TimeOfDay/DayOfWeek})}$ smooth functions consider the intra-day effects over the power load differently for each i-day-of-week, but those have already excluded the main intra-day effect regardless of the day of the week. This latter effect has already been captured in the $f^{(\text{TimeOfDay})}$ component. The same happens to $f^{(\text{Temperature/TimeOfDay})}$ and $f^{(\text{Temperature/DayOfYear})}$ components: those components capture the intra-day and annual seasonality effect of temperature considering that the main effect of temperature is already set up in the $f^{(\text{Temperature})}$ component[5].

The rest of the modeling activity follows a stepwise process in which the remaining variable is incorporated into the model at each iteration.

Third, to fit the model, package *mgcv* R was adopted using thin plate regression spline bases, once they can smooth any number of covariates, avoid defining knots, and have some optimal properties (view Section 4.1.5 and table 5.1 from [77] for more information). Although effective degrees of freedom are controlled by the degree of penalization selected during fitting, by Generalized Cross-Validation, the upper limit[6] on degrees of freedom is explicitly defined within each additive component [78].

Finally, the model metrics and residuals were analyzed. Consequently, if any insights emerge, that might lead to the repetition of this methodology to increase the accuracy of the model.

The following subsections split the rest of modeling activity into two subjects, one per type of explanatory variable, which improves the model accuracy.

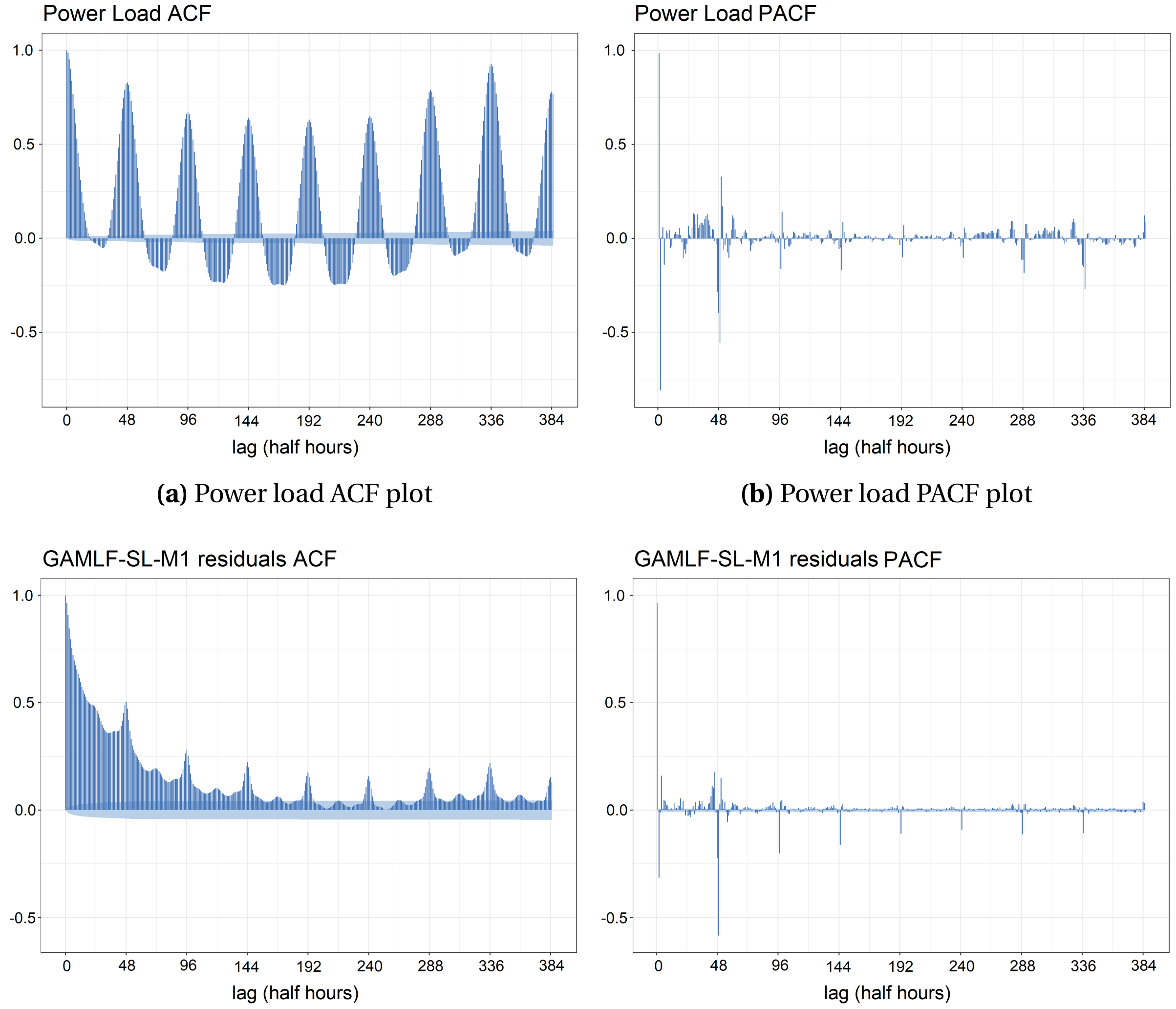
3.3.2 Lagged Load

The power load time series, as many other signals, show the presence of repeating patterns or periodic signals. To understand the autocorrelation of time series, Figures 3.3a and 3.3b exhibit the autocorrelation function (ACF) and the partial autocorrelation function (PACF).

[5] For reproducibility in R's *mgcv* package consider the formula `y ~ trend + s(timeOfDay, k=40) + ti(timeOfDay, by=dayOfWeek, k=20) + dayOfWeek + s(dayOfYear, k=40) + s(temperature, k=30) + ti(temperature, timeOfDay, k=7) + ti(temperature, dayOfYear, k=7)`

[6] Argument `k` which appears in the formula for the package *mgcv*.

In both plots, high coefficients rise at immediately lag values and, more important, at 24 hours lag, 2 days lag, 3 days lag, etc. This represents a daily seasonality. The amplitude of ACF of subsequently peaks decreases until 4 days lag but increases again until reaching a new maximum amplitude precisely at 7 days lag. This “scalloped” shape is due to the weekly seasonality.



(a) Power load ACF plot

(b) Power load PACF plot

(c) GAMLF-SL-M1 residuals ACF plot

(d) GAMLF-SL-M1 residuals PACF plot

Figure 3.3. ACF and PACF plots show the correlation of power load (3.3a and 3.3b) or GAMLF-SL-M1 residuals (3.3c and 3.3d) with a delayed copy of itself as a function of delay (8 days maximum). In both plots, high coefficients rise at immediately lag values and, more important, at lags of 48 (24 hours), 96 (2 days), 144 (3 days), and so on. The amplitude of ACF of subsequently peaks decreases until 4 days lag (192 half hours) but increases again until reaching a new maximum amplitude precisely at 7 days lag (lag 336). Notice that the coefficients (columns) lie beyond the light blue region as they signify strong statistical confidence.

The autocorrelation functions suggest which are the best lag periods as autoregressive terms to be included in the model structure. However, the final selection of autoregressive terms depends on the applicability of the model. The use of a lagged power load as an input into the model leads to the necessity that the data should be available right on time to predict. For example, considering the 24-hour lagged power load as input, the model would predict

Table 3.3. Taking into account 24 hours, 2 days, and 1 week as possible lagged power loads, the methodology is followed testing different combinations of candidate covariates added to the previous model, GAMLF-SL-M1. The function EWMA$_\alpha$ represents the exponential α-weighted moving average.

Model	Candidate covariates added to GAMLF-SL-M1
M2a	$f^{(\text{LagLoad1w})}\left(y_{t-336}\right)$
M2b	$f^{(\text{LagLoad48h})}\left(y_{t-96}\right)$
M2c	$f^{(\text{LagLoad48h})}\left(y_{t-96}\right)+f^{(\text{LagLoad1w})}\left(y_{t-336}\right)$
M2d	$f^{(\text{LagLoad24h})}\left(y_{t-48}\right)+f^{(\text{LagLoad48h})}\left(y_{t-96}\right)+f^{(\text{LagLoad1w})}\left(y_{t-336}\right)$
M2e	$f^{(\text{LagLoad24h})}\left(y_{t-48}\right)+f^{(\text{LagLoad48h})}\left(y_{t-96}\right)$
M2f	$f^{(\text{LagLoad24h})}\left(y_{t-48}\right)$
M2g	$f^{(\text{LagLoad24h})}\left(y_{t-48}\right)+f^{(\text{LagLoad1w})}\left(y_{t-336}\right)$
M2h	$f^{(\text{LagLoad24h})}\left(y_{t-48}\right)+f^{(\text{LagLoad1w})}\left(y_{t-336}\right)-\beta_1 x_t^{(\text{Trend})}$
M2i	$f^{(\text{EWMALagLoad24h})}\left(\text{EWMA}_{0.7}\left(y_{t-48},y_{t-49},y_{t-50},y_{t-51}\right)\right)$
M2j	$f^{(\text{EWMALagLoad1w})}\left(\text{EWMA}_{0.7}\left(y_{t-336},y_{t-337},y_{t-338},y_{t-339}\right)\right)$
M2k	M2i + M2j

a maximum horizon of 24 hours, considering that there is no delay in the availability of the power load time series.

However, it is important to note that the covariates already included in the GAMLF-SL-M1 structure could capture these different seasonalities. To ensure that new autoregressive terms will improve the accuracy of the model despite additional complexity, ACF and PACF are applied to the residuals of GAMLF-SL-M1 instead of the power load signal. Figures 3.3c and 3.3d suggest that even residuals manifest autocorrelation patterns and peaks are comparable with the ACF and PACF of the power load time series. Taking into account the structure of the previous model, GAMLF-SL-M1 (equation 3.7), different combinations of possible lagged power loads are added as candidate covariates. These models, M2a–M2k, as shown in Table 3.3, are evaluated with a one-year update cycle. The results are demonstrated in Table 3.5.

Finally, after evaluating, the final model, GAMLF-SL-M2, is based on the result of the M2g model and is defined as

$$\begin{aligned}
\hat{y}_t^{(\text{M2})} &= \beta_0+\beta_1 x_t^{(\text{Trend})}\\
&+f^{(\text{LagLoad24h})}\left(y_{t-48}\right)+f^{(\text{LagLoad1w})}\left(y_{t-336}\right)\\
&+f^{(\text{TimeOfDay})}\left(x_t^{(\text{TimeOfDay})}\right)\\
&+\sum_{i\in\{\text{Monday},\ldots,\text{Sunday}\}}\mathbf{1}_{\left(x_t^{\text{DayOfWeek}}=i\right)}\left(\beta_i+f_i^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)\right)\\
&+f^{(\text{DayOfYear})}\left(x_t^{(\text{DayOfYear})}\right)\\
&+f^{(\text{Temperature})}\left(x_t^{(\text{Temperature})}\right)\\
&+f^{(\text{Temperature/TimeOfDay})}\left(x_t^{(\text{Temperature})},x_t^{(\text{TimeOfDay})}\right)\\
&+f^{(\text{Temperature/DayOfYear})}\left(x_t^{(\text{Temperature})},x_t^{(\text{DayOfYear})}\right)\\
&+\epsilon_t
\end{aligned}\quad. \tag{3.8}$$

3.3.3 Calendar

The power load is largely dependent on human behavior, which in turn is conditioned by the calendar. The literature is full of various approaches to the calendar and time. For example:

- month (12 classes); months grouped by the four seasons (4 classes); months grouped into 7 types to distinguish the transitions between two adjacent seasons (7 classes); hot and cold days instead of months (2 classes);
- day of the week (7 classes); working and weekend days (2 classes); weekend, adjacent days to the weekend and other working days (3 classes); and other combinations of the 7 days;
- intraday steps (48 steps if half-hourly or 24 steps if hourly); or other combinations, for example: dawn, morning, lunch, afternoon, evening, and late night (6 classes);
- holidays treated separately; or treated as weekends; or even additionally modeling the surrounding days of a holiday apart from others.

The GAMLF-SL-M2 model already includes the 7 days of the week to discriminate the intraday power load into 7 typical curves, one per week. However, the residuals plotted over one year show patterns in specific days and periods. Figure 3.4 shows the patterns in the summer holidays, August, Christmas, New Year, and other surrounding public holidays. Figure 3.5 shows the challenging issue of modeling the surrounding days of a public day, in this case, Christmas Eve and the day after Christmas, for a model that already perceives Christmas day as a special day.

Taking into account the structure of the previous model, GAMLF-SL-M2 (equation 3.8), several adjustments are studied to cope with the existence of a holiday on the day the model forecasts, or the existence of a holiday on the day the autoregressive covariates uphold. These models, M3a–M3f, as shown in Table 3.4, are evaluated with a one-year update cycle. The results are demonstrated in Table 3.5. Let explain the adjustments:

- Model M3a takes into account a global value to increase or decrease prediction depending on whether the day being forecast is a public holiday;
- Model M3b adjusts the number of classes of set G from 7 to 9 of the component already defined in GAMLF-SL-M2, adding the class *HolidayOnWeekend* for public holidays that occur on the weekend and *Holiday* otherwise;
- Model M3c takes into account global values to increase or decrease prediction, specific for each public holiday (non-boolean off-day variable);
- Model M3d adds the autoregressive component *LagLoad48h* to the model that might be different depending on whether the previous day was a public holiday;
- Model M3e adjusts the autoregressive component *LagLoad24h* to the model perceive differences when it comes from a public holiday (which took place the day before);
- Model M3f adjusts the autoregressive component *LagLoad1w* to the model perceive differences when it comes from a public holiday (which happened one week ago);

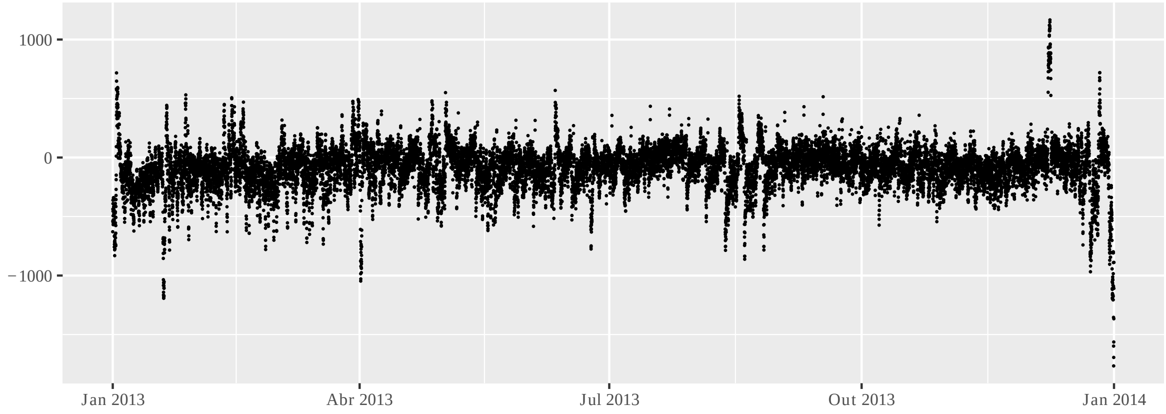


Figure 3.4. Residuals over a year are analyzed to check patterns and obtain insights. Note the residuals in summer holidays – August –, Christmas and New Year holidays, and surrounding other public holidays.

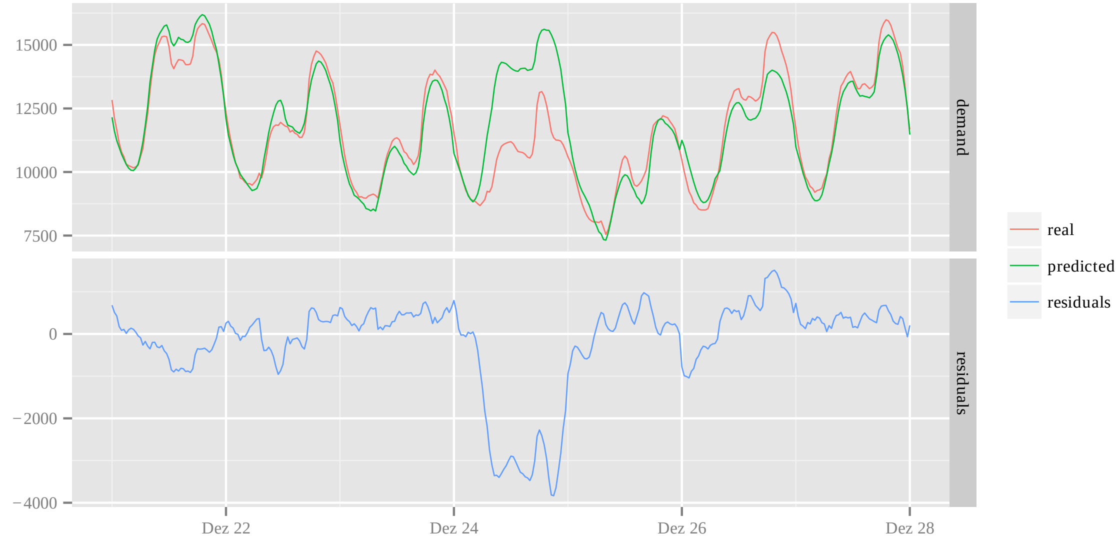


Figure 3.5. Although the model already perceives the Christmas day as a special day, the surrounding days – the Christmas Eve and the day after Christmas – continue to be a challenging issue. The figure plots the real (red) and predicted (green) values, as well as the difference (blue) between them. Note that 24th December is also adjacent to a weekend, that is, Christmas Eve is a Monday in this case.

Table 3.4. Considering the holiday existence on the day autoregressive covariates uphold (adjustments in M3c–e) or the holiday existence on the day model is forecasting for (adjustments in M3a–b and M3f), the methodology followed testing the different candidates.

Model	Candidate covariates added or adjusted to GAMLF-SL-M2
M3a	$\sum_{i\in\{\text{Yes},\text{No}\}} \mathbf{1}_{\left(x_t^{\text{isPublicHoliday}}=i\right)}\beta_i$
M3b	$\sum_{i\in G} \mathbf{1}_{\left(x_t^{\text{DayType}}=i\right)}\left(\beta_i + f_i^{(\text{TimeOfDay/DayType})}\left(x_t^{(\text{TimeOfDay})}\right)\right)$ where $G = \{\text{MondayNoHoliday}, \ldots, \text{SundayNoHoliday}, \text{Holiday}, \text{HolidayOnWeekend}\}$
M3c	M3b + $\sum_{i\in\{\text{NewYear},\cdots,\text{Christmas}\}} \mathbf{1}_{\left(x_t^{\text{PublicHoliday}}=i\right)}\beta_i$
M3d	M3c + $\sum_{i\in\{\text{Yes},\text{No}\}} \mathbf{1}_{\left(x_{t-48}^{\text{isPublicHoliday}}=i\right)}\left(\beta_i + f_i^{(\text{LagLoad48h})}\left(y_{t-96}\right)\right)$
M3e	M3d + $\sum_{i\in\{\text{Yes},\text{No}\}} \mathbf{1}_{\left(x_{t-48}^{\text{isPublicHoliday}}=i\right)}\left(\beta_i + f_i^{(\text{LagLoad24h})}\left(y_{t-48}\right)\right)$
M3f	M3e + $\sum_{i\in\{\text{Yes},\text{No}\}} \mathbf{1}_{\left(x_{t-336}^{\text{isPublicHoliday}}=i\right)}\left(\beta_i + f_i^{(\text{LagLoad1w})}\left(y_{t-336}\right)\right)$

Finally, after evaluating, the final model, GAMLF-SL-M3, is based on the results of the M3c model and is defined as

$$
\begin{aligned}
\hat{y}_t^{(\text{M3})} &= \beta_0 + \beta_1 x_t^{(\text{Trend})} \\
&+ f^{(\text{LagLoad24h})}\left(y_{t-48}\right) + f^{(\text{LagLoad1w})}\left(y_{t-336}\right) \\
&+ f^{(\text{TimeOfDay})}\left(x_t^{(\text{TimeOfDay})}\right) \\
&+ \sum_{i\in G} \mathbf{1}_{\left(x_t^{\text{DayType}}=i\right)}\left(\beta_i + f_i^{(\text{TimeOfDay/DayType})}\left(x_t^{(\text{TimeOfDay})}\right)\right) \\
&+ \sum_{j\in\{\text{NewYear},\cdots,\text{Christmas}\}} \mathbf{1}_{\left(x_t^{\text{PublicHoliday}}=j\right)}\beta_j \\
&+ f^{(\text{DayOfYear})}\left(x_t^{(\text{DayOfYear})}\right) \\
&+ f^{(\text{Temperature})}\left(x_t^{(\text{Temperature})}\right) \\
&+ f^{(\text{Temperature/TimeOfDay})}\left(x_t^{(\text{Temperature})}, x_t^{(\text{TimeOfDay})}\right) \\
&+ f^{(\text{Temperature/DayOfYear})}\left(x_t^{(\text{Temperature})}, x_t^{(\text{DayOfYear})}\right) \\
&+ \epsilon_t
\end{aligned}
\tag{3.9}
$$

where $G = \{\text{MondayNoHoliday}, \ldots, \text{SundayNoHoliday}, \text{Holiday}, \text{HolidayOnWeekend}\}$.

The low improvement of M3d, M3e, and M3f models accuracy compared to M3c's does not justify their adoption due to complexity increase.

3.3.4 Results

In this section, the generalized additive model-based load forecasting – system-level model, GAMLF-SL, is evaluated. The same dataset and the procedures to compute the results are followed as in Section 3.2.2.

The results for GAMLF-SL are shown in Tables 3.6 and 3.7 for the same update cycles. As a first result, it is obvious that GAMLF-SL-M1 achieves a better accuracy compared to the results

Table 3.5. Analysis of candidates for variables testing different adjustments of previous models evaluated with one year update cycle.

Model	**MAE** (MW)	**MAPE** (%)	**RMSE** (MW)	**NRMSE** (%)	$\mathbf{R^2}$	**MASE**
M2a	190.25	3.43	286.96	4.98	0.923	0.419
M2b	183.32	3.32	280.56	4.88	0.926	0.404
M2c	168.06	3.03	263.77	4.59	0.932	0.371
M2d	140.80	**2.53**	229.49	**3.99**	0.948	**0.311**
M2e	148.64	2.68	237.79	4.14	0.945	0.328
M2f	149.59	2.70	238.01	4.14	0.946	0.330
M2g	**140.77**	**2.53**	**229.16**	**3.99**	0.948	**0.311**
M2h	143.82	2.54	229.44	**3.99**	**0.949**	0.317
M2i	151.36	2.73	239.91	4.17	0.945	0.334
M2j	191.48	3.45	287.30	5.00	0.923	0.422
M2k	142.32	2.55	230.85	4.02	0.948	0.314
M3a	137.31	2.47	210.89	3.67	0.956	0.303
M3b	130.31	2.32	197.56	3.44	0.962	0.287
M3c	126.97	2.26	191.08	3.32	**0.965**	0.280
M3d	125.84	2.24	192.26	3.35	0.964	0.278
M3e	125.34	2.23	190.68	3.32	0.964	0.276
M3f	**124.12**	**2.21**	**189.56**	**3.30**	**0.965**	**0.274**

of the benchmark model, GLMLF-B. Taking into account the same or similar variables, the GAM technique achieves better accuracy with simplified interpretability (discussed later). Moreover, the systematic approach iteratively enhances the GAM-based regression models by introducing new synthetic explanatory variables based on domain knowledge. GAMLF-SL-M2 adds autoregressive components to the previous model, and GAMLF-SL-M3 in turn adjusts and adds calendar features. When comparing the benchmark model, GLMLF-B, and the enhanced GAMLF-SL-M3 model, MAPE, MAE, RMSE and NRMSE were reduced by 42% to 47% in all update cycle scenarios. Furthermore, as the median and average accuracy improves, the days with higher errors also improve in terms of error amplitude. This is demonstrated by the decrease in spread of the box plots exhibited in Figure 3.6. The MAPE and MAE maximums, meaning the days with higher errors, decrease across the four models cross-validated with 365 folds, each fold representing one day of testing. Figure 3.7 shows the improvement on special days, such as warmer summer days in June, July, and August, as well as on weekends and holidays. However, the model GAMLF-SL-M3 introduced a higher error on public holidays that occur on weekends compared to other models.

With respect to interpretability, the internals of the model are easy to interpret. Besides the parameters β, the spline functions f, which are also estimated during the fitting step, are easy to understand and familiar. The fact that experts see their domain knowledge “recognized” by the model reinforces acceptance and adoption. Figure 3.8 shows the main effect of the intra-day pattern (3.8a), as well as the interaction of the same effect for each day of the week after the main effect being appropriately excluded (3.8b-3.8i). Note that the model adds a lower load baseline for weekends compared to working days (Figure 3.8b). Consequently, to maintain the same levels of power load during the very early morning (0 a.m. to 6 a.m.), as on any other ordinary day, f_{Saturday} (3.8h) and f_{Sunday} (3.8i) display, for that period, higher

Table 3.6. GAMLF-SL results calculated over the time span from 2016 to 2019 using time series cross-validation with a fixed 3-year window for training and the next 1 year (1-fold), 2 weeks (26-fold), 1 week (52-fold) or 1 day (365-fold) of testing data. The best results are in bold. MASE was calculated with m-steps equal to 52 weeks, 2 weeks, 1 week, and 1 day, respectively. All metrics are the mean calculated from cross-validation folds.

	Model	MAE (MW)	MAPE (%)	RMSE (MW)	NRMSE (%)	R^2	MASE
1 year	M1	227	4.09	323	5.61	0.909	0.501
	M2	141	2.53	229	3.99	0.948	0.311
	M3	**127**	**2.26**	**191**	**3.32**	**0.965**	**0.280**
2 weeks	M1	202	3.59	276	4.81	0.927	0.699
	M2	138	2.47	198	3.48	0.944	0.468
	M3	**124**	**2.20**	**172**	**3.01**	**0.963**	**0.433**
1 week	M1	200	3.56	265	4.64	0.933	0.907
	M2	138	2.46	190	3.35	0.946	0.603
	M3	**124**	**2.19**	**166**	**2.91**	**0.964**	**0.568**
1 day	M1	194	3.46	234	4.16	0.957	1.142
	M2	137	2.44	166	2.97	0.967	0.686
	M3	**123**	**2.18**	**148**	**2.64**	**0.977**	**0.653**

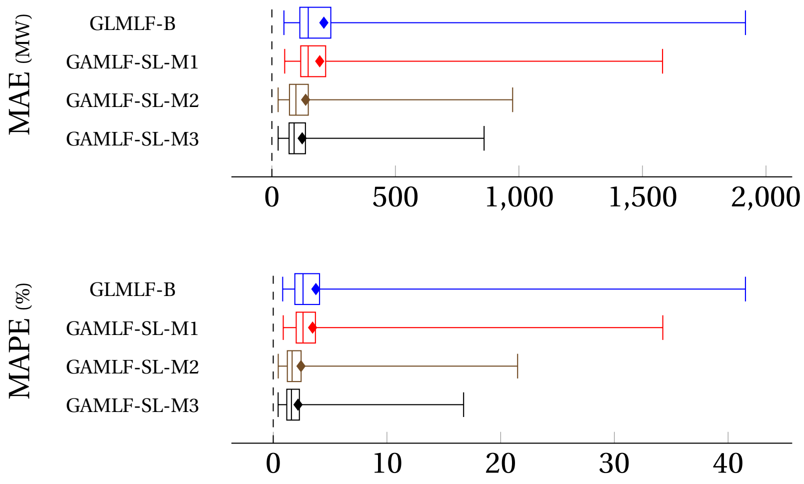
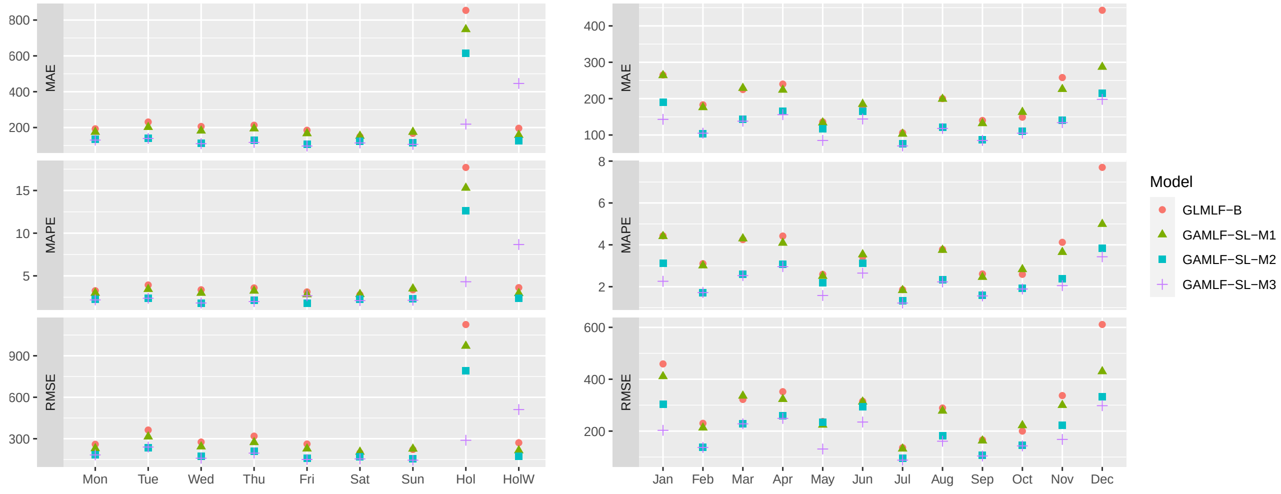


Figure 3.6. Quantile box plots of MAPE and MAE for the four models cross-validated with 365 folds (1 day update cycle). Not only do the median and average accuracy improve, but days with higher errors are also predicted more accurately.

Table 3.7. GAMLF-SL results calculated over the time span from 2016 and 2019 using time-series cross-validation with a 3-years fixed window for training and the next 1 year (1 fold), 2 weeks (26 folds), 1 week (52 folds) or 1 day (365 folds) of testing data. The MASE was calculated with m-steps equal to 52 weeks, 2 weeks, 1 week and 1 day respectively. The minimum, maximum, mean, and the first, second (median) and third quartiles are calculated for all metrics on each cross-validation fold.

	Model	1 year	2 weeks						1 week						1 day					
		Median	Min	1Q	Median	3Q	Max	Mean	Min	1Q	Median	3Q	Max	Mean	Min	1Q	Median	3Q	Max	Mean
MAE (MW)	GLMLF-B	235.25	99.33	133.99	185.46	246.94	606.27	217.75	76.19	131.78	174.26	257.90	760.52	214.64	48.77	113.11	147.17	238.50	1916.92	210.46
	GAMLF-SL-M1	226.86	96.10	136.54	192.13	230.32	241.92	201.63	83.48	128.19	170.11	242.41	489.81	199.88	51.85	116.67	146.84	217.61	1581.24	193.66
	GAMLF-SL-M2	140.77	64.17	91.92	105.84	173.89	317.19	137.89	60.96	86.36	110.16	155.70	390.15	137.67	25.32	71.25	97.28	147.37	974.48	136.63
	GAMLF-SL-M3	126.97	60.48	90.41	115.65	139.73	263.02	123.91	57.13	83.67	110.15	135.88	329.43	123.62	25.38	69.66	90.01	135.80	858.93	122.69
MAPE (%)	GLMLF-B	4.23	1.75	2.44	3.25	4.74	11.48	3.87	1.40	2.38	2.84	4.85	15.18	3.82	0.83	1.90	2.62	4.07	41.53	3.75
	GAMLF-SL-M1	4.09	1.73	2.47	3.23	4.13	7.90	3.59	1.50	2.23	2.93	4.14	9.77	3.56	0.88	2.02	2.62	3.71	34.26	3.46
	GAMLF-SL-M2	2.53	1.15	1.57	1.91	3.29	6.06	2.47	1.11	1.54	1.82	2.56	7.66	2.46	0.44	1.24	1.66	2.45	21.49	2.44
	GAMLF-SL-M3	2.26	1.07	1.58	1.93	2.53	4.83	2.20	1.02	1.49	1.86	2.23	6.57	2.19	0.43	1.19	1.60	2.30	16.74	2.18
RMSE (MW)	GLMLF-B	353.12	123.09	170.61	259.29	390.05	814.17	296.99	95.06	166.22	215.53	386.01	979.36	283.73	60.54	138.97	181.33	287.26	2049.36	249.74
	GAMLF-SL-M1	322.65	122.33	175.85	254.70	336.20	585.08	275.59	108.50	160.03	218.70	354.80	738.20	264.56	66.88	142.60	178.94	255.63	1747.64	233.66
	GAMLF-SL-M2	229.16	84.06	118.43	137.38	287.37	461.68	198.11	76.81	105.26	140.27	196.57	549.17	189.97	35.52	88.37	116.09	175.75	1093.63	165.63
	GAMLF-SL-M3	191.08	76.31	117.85	149.17	190.02	374.88	172.28	74.28	103.50	143.08	174.43	437.58	165.82	32.88	85.90	112.40	160.49	982.90	148.25
NRMSE (%)	GLMLF-B	6.14	2.20	3.05	4.13	6.58	14.43	5.17	1.69	2.99	3.71	6.42	18.46	4.96	1.02	2.40	3.20	4.78	43.29	4.44
	GAMLF-SL-M1	5.61	2.18	3.14	4.09	6.17	10.37	4.81	1.98	2.84	3.61	5.88	12.71	4.64	1.14	2.48	3.14	4.46	36.91	4.16
	GAMLF-SL-M2	3.99	1.47	2.09	2.41	5.27	8.20	3.48	1.40	1.87	2.30	3.43	9.48	3.35	0.61	1.50	2.05	2.97	23.28	2.97
	GAMLF-SL-M3	3.32	1.33	2.09	2.56	3.49	6.66	3.01	1.32	1.83	2.37	3.07	8.32	2.91	0.56	1.50	1.95	2.83	19.39	2.64
R^2	GLMLF-B	0.888	0.597	0.877	0.963	0.974	0.985	0.918	0.614	0.943	0.971	0.980	0.988	0.929	0.342	0.965	0.980	0.988	0.997	0.958
	GAMLF-SL-M1	0.909	0.723	0.870	0.963	0.975	0.985	0.927	0.637	0.946	0.971	0.978	0.988	0.933	0.312	0.964	0.977	0.986	0.996	0.957
	GAMLF-SL-M2	0.948	0.763	0.903	0.980	0.986	0.993	0.944	0.696	0.964	0.983	0.988	0.993	0.946	0.189	0.983	0.991	0.995	0.998	0.967
	GAMLF-SL-M3	0.965	0.851	0.964	0.978	0.983	0.994	0.963	0.717	0.966	0.981	0.989	0.994	0.964	0.444	0.984	0.991	0.995	0.998	0.977
MASE	GLMLF-B	0.519	0.323	0.486	0.661	0.902	1.916	0.742	0.220	0.643	0.850	1.263	3.212	0.970	0.063	0.263	0.633	1.718	9.193	1.207
	GAMLF-SL-M1	0.501	0.316	0.485	0.623	0.870	1.274	0.699	0.241	0.619	0.833	1.197	2.188	0.907	0.073	0.254	0.621	1.518	9.300	1.142
	GAMLF-SL-M2	0.311	0.213	0.347	0.425	0.590	0.962	0.468	0.264	0.427	0.561	0.727	1.212	0.603	0.056	0.171	0.410	1.001	4.298	0.686
	GAMLF-SL-M3	0.280	0.170	0.316	0.399	0.514	0.828	0.433	0.232	0.344	0.539	0.697	1.620	0.568	0.063	0.159	0.368	0.914	4.894	0.653



(a) Accuracy metrics by day type

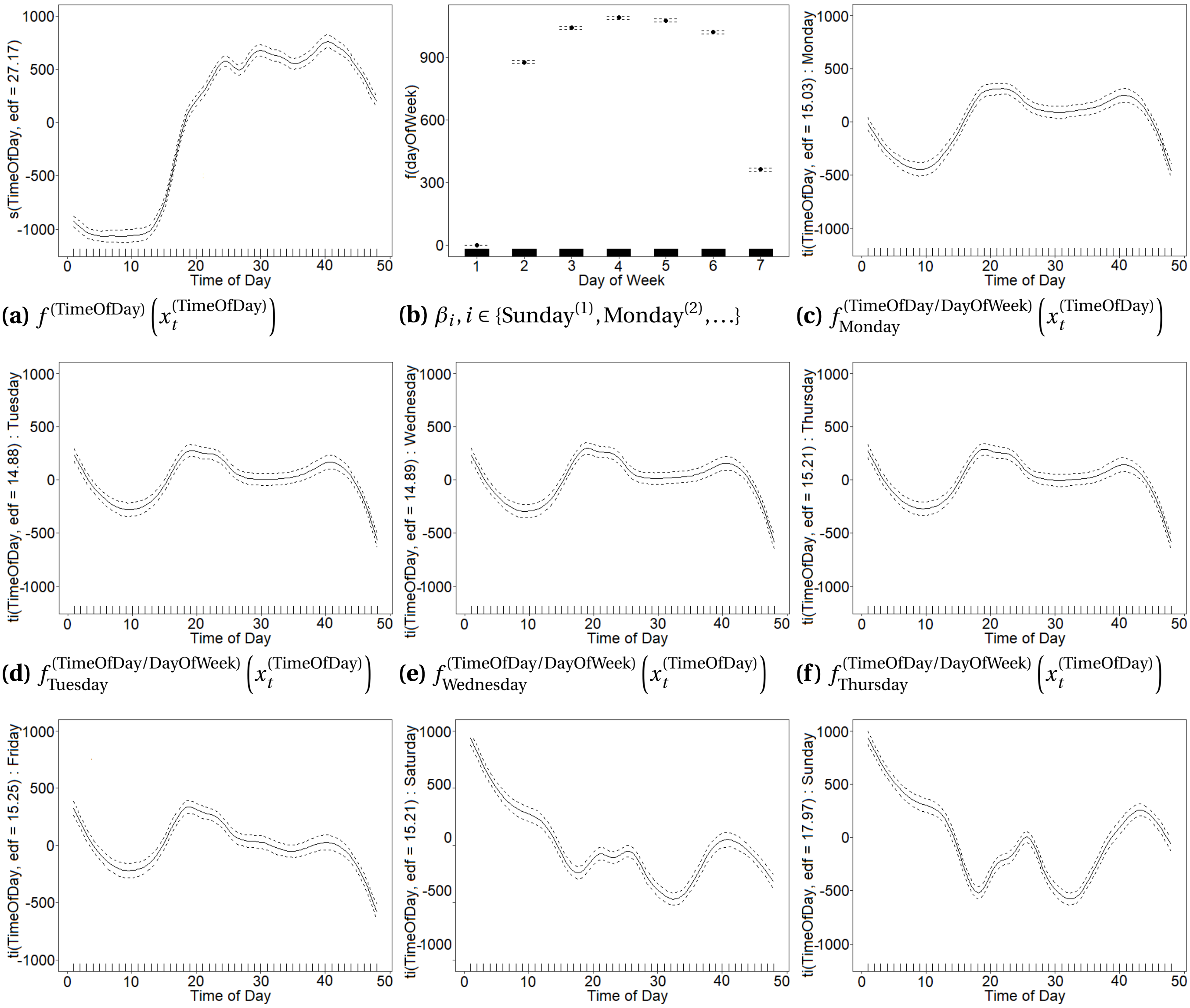
(b) Accuracy metrics by month

Figure 3.7. MAE, MAPE, and RMSE by month or day type calculated for the four models cross-validated with 365 folds (1 day update cycle). The day type "Hol" means holiday and "HolW" means holiday on a weekend. The day types are disjointed subgroups.

responses compared to other spline functions (3.8c-3.8g). Thus, this compensates for the low load baseline added by β_{Saturday} and β_{Sunday} (3.8b). On the other hand, the peaks and valleys of those two weekend spline functions are different from those of the working days. Even the number of effective degrees of freedom is higher for those two days due to the more complex structure required to reshape the baseline $f^{(\text{TimeOfDay})}$. Note that the baseline multiplied by β_i, which is always added, gives a more approximate shape to the load curve of the working day.

Furthermore, β_1, which results from the linear component of the time series, and $f^{(\text{DayOfYear})}$, which emerges from the annual pattern, are shown in Figure 3.9. Note the pronounced valley shape at the beginning and end of the year, as well as in the summer leave (August): they suggest a trend as lower electricity consumption. The effect of temperature on the power load is highlighted in the graphs in Figure 3.10. The main effect of temperature (3.10a) is aligned with the scattered plot in Figure 3.2. It can be seen from 3.10b that low temperatures during the early morning (0 am to 6 am) have the opposite effect to the same low temperature during the evening. During the evening, low temperatures reinforce the general pattern (3.2), that is, increase the power load. On the contrary, during the early morning, low temperatures shrink the effect of that same general pattern. Another insight is that warm winters have the effect of shrinking load predictions due to a reduction in heat demand (3.10c).

More insights could be derived from the internals of the fitted model, the point is: the model proposes a highly interpretable internal structure.



(a) $f^{(\text{TimeOfDay})}\left(x_t^{(\text{TimeOfDay})}\right)$

(b) $\beta_i, i \in \{\text{Sunday}^{(1)}, \text{Monday}^{(2)}, \ldots\}$

(c) $f_{\text{Monday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

(d) $f_{\text{Tuesday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

(e) $f_{\text{Wednesday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

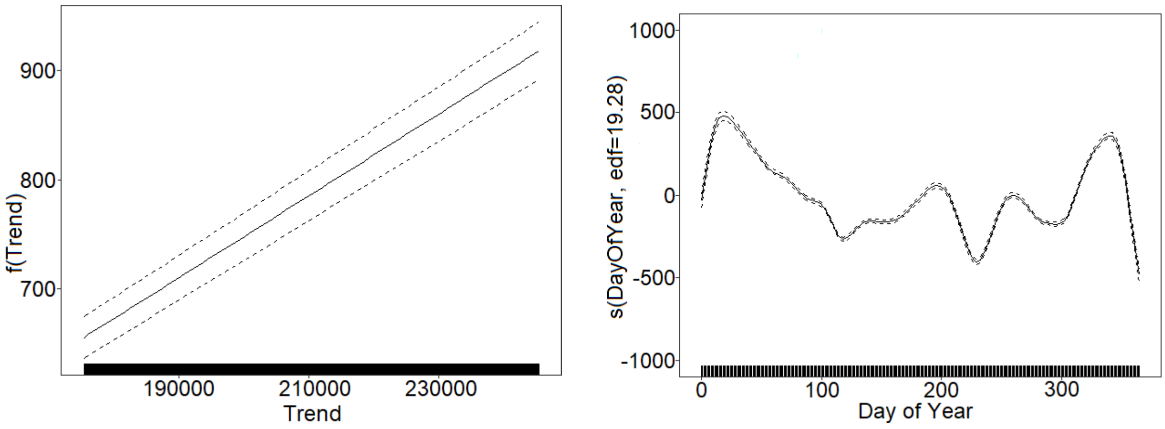
(f) $f_{\text{Thursday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

(g) $f_{\text{Friday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

(h) $f_{\text{Saturday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

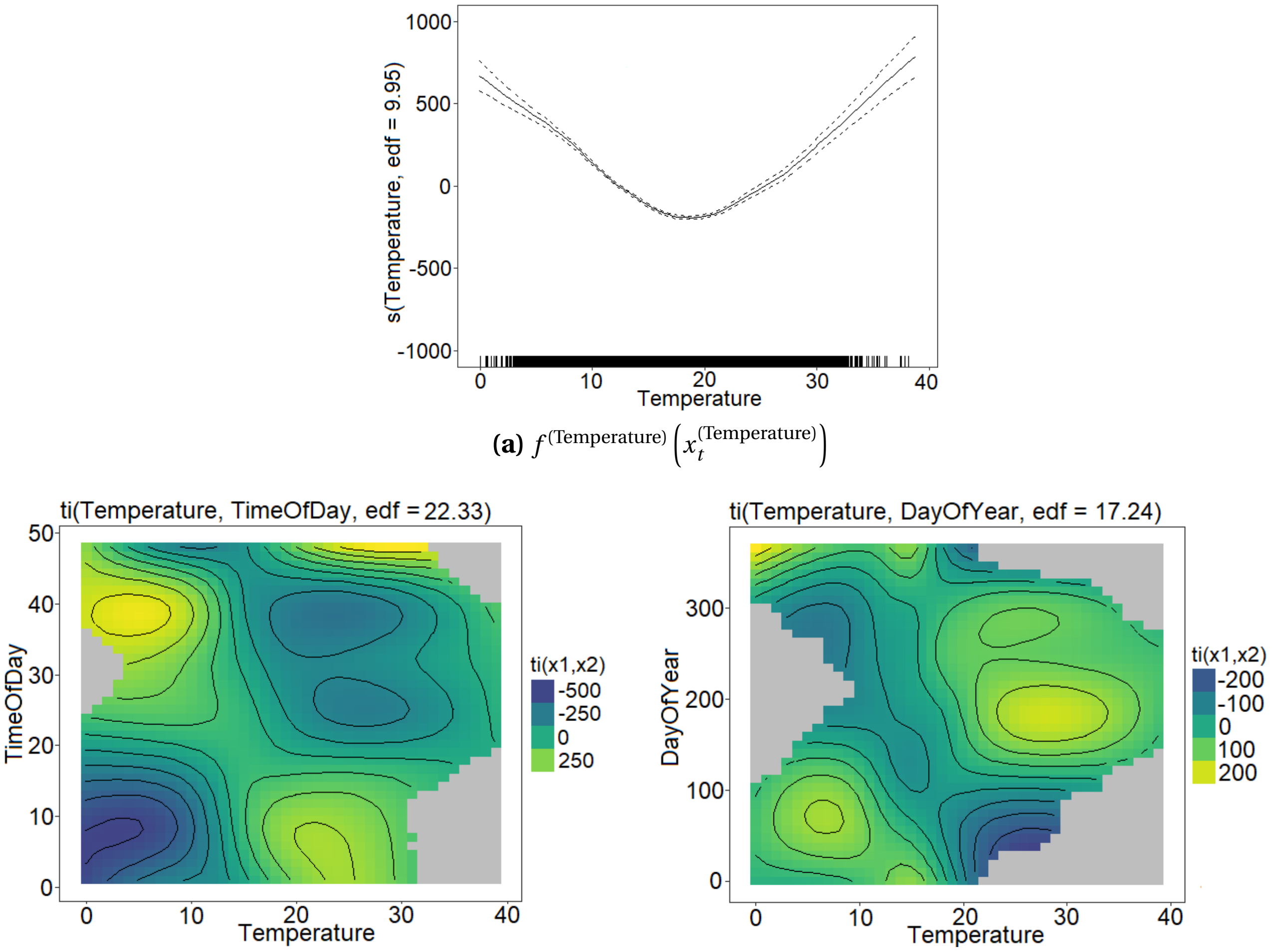
(i) $f_{\text{Sunday}}^{(\text{TimeOfDay/DayOfWeek})}\left(x_t^{(\text{TimeOfDay})}\right)$

Figure 3.8. The effects of intra-daily and weekly patterns on the power load are recognized by the parameters β and spline functions f which were estimated during the fitting step of the GAMLF-SL-M1 model. The figure (a) exhibits the main effect of intra-day pattern independently of the day of week, while figures (c-i) exhibit the effect of intra-day pattern for each day of the week after the main effect being appropriately excluded.



(a) $\beta_1 x_t^{(\text{Trend})}$ **(b)** $f^{(\text{DayOfYear})}\left(x_t^{(\text{DayOfYear})}\right)$

Figure 3.9. The linear component and the annual pattern recognized by the β_1 parameter and the $f^{(\text{DayOfYear})}$ spline function, which are estimated during the fitting step of model GAMLF-SL-M1 model.



(a) $f^{(\text{Temperature})}\left(x_t^{(\text{Temperature})}\right)$

(b) $f^{(\text{Temperature/TimeOfDay})}\left(x_t^{(\text{Temperature})}, x_t^{(\text{TimeOfDay})}\right)$ **(c)** $f^{(\text{Temperature/DayOfYear})}\left(x_t^{(\text{Temperature})}, x_t^{(\text{DayOfYear})}\right)$

Figure 3.10. The effects of temperature over the power load recognised by the spline functions f which were estimated during the fitting step of model GAMLF-SL-M1. Figure (b) and (c) represent the bivariate effect of the temperature and the time of the day (b) or the day of the year (c).

3.4 Gradient Boosting Model – GBMLF-SL

In the space of ensemble methods, gradient boosting machines (GBM) have been a prominent technique used in a diverse number of machine learning and data mining challenges with considerable success [79]. The main idea of boosting is to add new learners to the ensemble formation sequentially. At each step, a new weak base-learner model is trained with respect to the error of the whole ensemble. Thus, the new base-learner formation is correlated with the negative gradient of the loss function associated with the whole ensemble [80, 81].

The flexibility makes GBM highly customizable to diverse machine-learning tasks. It has been considered to energy domain data-driven tasks, including electricity load forecasting [82–84]. Hence, GBM's success in disparate domain encourages its assessment and comparison with the models of previous sections.

The gradient boosting machine load forecasting — GBMLF-SL — is also a regression model for system level, as previous models already evaluated. The assessment is conducted using the same performance metrics and the same features which model GAMLF-SL-M3c used, though an adequate pre-processing is needed to apply GMB. Three different base learners and explored: (i) tree-based model, (ii) linear functions with L_1 and L_2 regularization, and (iii) component-wise smoothing splines.

3.4.1 Model

In addition to the dependent variable $\mathbf{y}$ already defined, consider the vector feature $\mathbf{x}_t$ that includes the same set of variables/information that the previous models used:

$$\mathbf{x}_t = \begin{bmatrix} x_t^{(\text{Trend})} & x_t^{(\text{DayType})} & x_t^{(\text{PublicHoliday})} & x_t^{(\text{DayOfYear})} & x_t^{(\text{TimeOfDay})} \\ x_t^{(\text{LagLoad24h})} & x_t^{(\text{LagLoad1w})} & x_t^{(\text{Temperature})} & x_t^{2\,(\text{Temperature})} & x_t^{3\,(\text{Temperature})} \end{bmatrix}^\top . \tag{3.10}$$

Here, the temperature data are explicitly unfold to three exponential versions (the temperature raised to the power of 1, 2 and 3), considering equal rationale when the same three components were used in the benchmark model (Section 3.2). The 1-day and 1-week lagged loaded are also explicitly included in the vector feature. Note that $x_t^{(\text{LagLoad24h})} \equiv y_{t-48}$, $x_t^{(\text{LagLoad1w})} \equiv y_{t-336}$, and $x_t^{(\text{DayType})} \in \{\text{MondayNoHoliday}, \ldots, \text{SundayNoHoliday}, \text{Holiday}, \text{HolidayOnWeekend}\}$. Remark that the base learners set a predefined internal structure unlike GAM in which the components interaction are explicitly modeled and constrained into the additive structure.

Additionally, categorical features are encoded into binary variables once the main GBM implementations are unable to operate on label data directly. The categorical features, $x_t^{(\text{DayType})}$ and $x_t^{(\text{PublicHoliday})}$, which do not establish a natural ordering between categories, are one-hot encoded, that is, a new binary variable is added for each unique category level.

Particularly, for the linear base learner with L_1 and L_2 regularization more sensible to different scales between features, the numerical variables are also pre-processed through normalization. Each numerical variable is individually scaled and centered in order to have a standard deviation of one and a mean of zero.

The Extreme Gradient Boosting algorithm (XGboost) was used with a hyperparameter exhaustive grid search in accordance with Table 3.8.

Table 3.8. Hyperparameter space search.

Model	Hyperparameter	Grid Search	Optimized Value
Tree-based	eval_metric	RMSE	RMSE
	n_rounds	50, 150, 200, $\cdots$, 10 000	1400
	eta	0.005, 0.01, 0.02, 0.05	0.02
	max_depth	1, 2, 3, $\cdots$, 10	6
	gamma	0, 5, 10, 15	0
	colsample_bytree	0.5, 0.6, 0.7	0.6
	min_child_weight	1, 3, 5, 7	1
	subsample	0.3, 0.4, 0.5, 0.6, 0.7	0.6
Linear-based	eval_metric	RMSE	RMSE
	n_rounds	50, 150, $\cdots$, 1000	250
	eta	0.005, 0.01, 0.02, 0.1, 0.2	0.005
	alpha (for L_1)	0.001, 0.01, 0.1, 1, 10, 100, 1000	1
	lambda (for L_2)	0.001, 0.01, 0.1, 1, 10, 100, 1000	0.001

3.4.2 Results

The results for GBMLF-SL indicate no improvements in accuracy when comparing the RMSE with the previous model, GAMLF-SL. The 199MW RMSE achieved compares to the 191MW RMSE of GAMLF-SL-M3, the latter better. Figure 3.11 shows the RMSE on the testing dataset throughout the boosting iterations of the algorithm. Each line progression and differences among plots suggest the sensibility to the hyperparameters during its optimization search. From the plots, it is possible to figure that the GBM outfits for some hyperparameters, those lines whose RMSE start to rise after a specific number of iterations.

Given the same data modeling, even with some tweaks to accommodate the characteristics of the gradient boosting machine (GBM), it achieves the same accuracy, but with two disadvantages to be noted.

GBM has many hyperparameters and they need to be optimized by an outside optimization algorithm that figures the hyperparameters based on minimization of loss. That takes computing time and effort to choose the optimization algorithm and define the grid search, though it is easily parallelizable.

Furthermore, GBM ensemble structure are notably less interpretable compared to GAM, and the purposed evaluation of the models ensures the qualitative balance between their accuracy and interpretability. Although weak base learners are known as highly interpretable as linear functions or small-sized trees, to achieve the same levels of accuracy, GBM holds thousands of linear functions or trees weightily combined to formulate the ensemble, losing the interpretability from the domain perspective. Nevertheless, model-agnostic interpretation techniques for machine learning models such as partial dependence plots (PDP), permutation feature importance (PFI) and Shapley values provide insightful model interpretations [85–87] if well used [88].

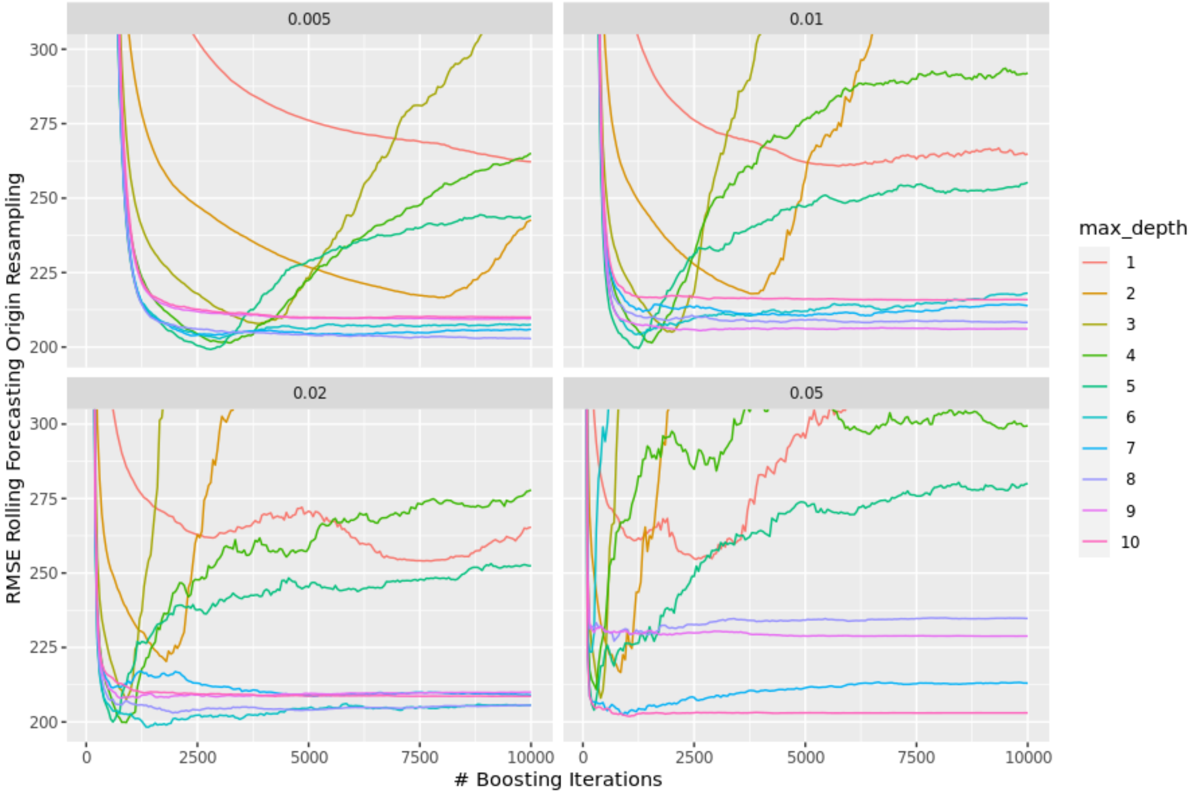


Figure 3.11. RMSE on the testing dataset throughout the boosting iterations of GBM. Each plot considers a different value for the *eta* hyperparameter (step size shrinkage used in the update to prevent overfitting). Each line considers the maximum depth of tree-based learner (increasing this value will make the model more complex). The minimum loss, 199MW of RMSE, is found with the optimized hyperparameters figured in Table 3.8.

3.5 Ensemble Model – GAMLF-SLE

The GAM based load forecasting, GAMLF-SLE, is a system-level regression model that results from the ensemble of a set of learners based on the previous sections. Ensemble methods can be applied to national load forecasting to improve accuracy. There are periods like the summer break in August and days around public holidays that are particularly challenging in forecasting tasks.

Ensemble methods get a collection of learners and combine their outcomes to make an overall decision. The rationale behind this is due to the fact that different learners, often referred to as experts, might be more accurate on specific subproblem, period time, or category. Therefore, a mechanism to build different learners that can capture specific information or patterns in the data must be defined.

3.5.1 Weighted Majority Algorithm

The Weighted Majority Algorithm (WM) is a simple and effective method, based on weighted voting, to build up a compound algorithm from the outputs of some pool of known algorithms where at least one of them will perform well [89].

WM learning proceeds in a sequence of trials. At each trial, the algorithm receives the outcomes of the predictors and uses them to process its own outcome considering the accuracy that the WM algorithm knows about each predictor at that moment. This accuracy knowledge is translated as weights, one for each predictor, and updated at the end of the trial according to the difference between each predictor outcome and the actual value, the label, which the WM algorithm receives delayed. WM evaluates such algorithms according to how many mistakes they make along the sequence.

3.5.2 Weighted Majority Algorithm Reviewed

The original Weighted Majority Algorithm is reviewed to cope with specific requirements of our scenario. Some modifications were also taken into account and implemented in other studies, but, to our best knowledge, there are some changes that are not found in the research area. The WM's specific requirements for our scenario are:

- be able to support numerical and unbounded outcomes from predictors;
- be able to cope with non-available outcomes of a subset of predictors and update the weights in accordance;
- be able to receive the label with a constant time delay and hold back the learning process;
- be able to learn in independent and predefined spans, called regimes, where each one is a discontinuous and disjoint temporal period.

The original paper discusses the case in which the predictions and labels range between 0 and 1. The algorithm variant is known as the Weighted Majority – Continuous version (WMC). Since our predictors' outcomes – power load forecasts – are numerical and unbounded, an unreachable maximum demand value should be defined to set an interval that is convertible to the working range of the algorithm. Data analysis indicates that a power of 10 000 MW appears to be unreachable; therefore, a linear mapping f from $[0, 10\,000]$ to $[0, 1]$ is applied to predictors' outcomes before entering the WMC algorithm, and whose output, in turn, is mapped back through f^{-1}, such that:

$$f\colon \left[0, 10000\right] \in \mathbb{R} \to \left[0,1\right] \in \mathbb{R}, \quad y \mapsto \frac{y}{10000}\,. \tag{3.11}$$

In our scenario, the predictors may not forecast in specific trials due to their own model design process. Indeed, some predictors will be designed and calibrated exclusively for specific periods such as summer, Christmas, New Year, weekends, public holidays, and off-days. So those models are not able to predict outside the period for which they are built up and consequently the WMC algorithm receives occasionally a non-available outcome from them. Consequently, the WMC algorithm needs to adopt a strategy to update the weight of each predictor when receiving a mix set of non-available and actual predictions. Several strategies are available, such as: (i) do not update the weights of predictors whose outcome is not available, (ii) update through a decreasing constant, or (iii) update in accordance with the worst, (iv) the best, or (v) the average of available predictions.

In the original WMC, for each forecaster i, the weight ω_i is updated in step $t+1$ by multiplying itself by the calculated value α_i, such that $0 < \alpha_i \leq 1$, i.e.,

$$\omega_i^{(t+1)} = \omega_i^{(t)} \alpha_i\,. \tag{3.12}$$

Since all weights are initialized as positive values, each update step results in lower weights. α_i is calculated for each prediction $\hat{y}_i \in [0,1]$ as

$$\alpha_i = \exp^{-\eta \cdot |\hat{y}_i - y|} \tag{3.13}$$

where $\eta > 0$ is a learning parameter and $y \in [0,1]$ is the label, the real value.

In this proposed algorithm, the adopted strategy updates the weights whose predictors do not forecast, according to the incur loss $\ell_{\text{abs}} = \left|\hat{y} - y\right|$ of the WMC algorithm outcome $\hat{y}$. Thus, these predictors do not benefit because the lack of outcome is negatively reflected in their weights, nor are they jeopardized since their weights are updated approximately at the same rate as the average, allowing them to contribute to the following trials without being more relevant just because there is no outcome.

Furthermore, the algorithm receives the label $y^{(t)}$ within a constant time delay δ. As the predictors are forecasting the load, for instance, 24 hours ahead (48 half-hours), the algorithm only gets the related label 48 trials later. Therefore, in the 49th trial, the WMC gets the first label and consequently can update the weights based on the forecasts calculated 48 trials ago. So, Equations 3.12 and 3.13 become

$$\omega_i^{(t+1)} = \omega_i^{(t)} \exp^{-\eta \cdot \left|\hat{y}_i^{(t-\delta)} - y^{(t-\delta)}\right|}\,. \tag{3.14}$$

Finally, different weights should be learned for different days. Due to the fact that predictors are designed and calibrated with different goals, some are more accurate to specific temporal intervals than the general predictor. For example, a predictor that models the power load exclusively during the summer break is empirically more accurate in that specific period than a model that models the whole year. To allow WMC to have different preferences depending on the actual time t, the concept of regime is introduced. Regime r is a discontinuous and disjoint temporal period in which the WMC algorithm learns and predicts. It uses distinct weights $w_{i,r}$ to maintain internal information for different periods, as well as distinct learning parameters η_r.

Algorithm 1 is the variant of the original WMC algorithm that was changed to meet the identified requirements.

Algorithm 1 Weighted Majority Algorithm – Continuous Reviewed Version

```
Let $N$ be the number of forecasters, and $R$ the number of regimes.
Choose the learning parameters $\eta_r > 0$ that would be used in the weight update rule.
Choose the time delay parameter $\delta > 0$.

Initiate weights $\omega^{(1)}_{i,r} \leftarrow 1 \ \forall_{i \in 1 \ldots N} \ \forall_{r \in 1 \ldots R}$
for $t \leftarrow 1 \ldots T$ do
    Receive instance $x \in \mathscr{X}$ of regime $r \in 1 \ldots R$
    Receive forecasts $y_1 \ldots y_N \in [0,1]^N$
    Predict $\hat{y} \leftarrow \frac{\sum_{i \in 1 \ldots N \wedge y_i \text{exists}} \omega_{i,r} y_i}{\sum_{i \in 1 \ldots N \wedge y_i \text{exists}} \omega_{i,r}} \in [0,1]$
    if $t > \delta$ then
        Let $r'$ be the regime at time $t - \delta$ that is $r' \equiv r^{(t-\delta)}$
        Let $\hat{y}'$ be the prediction at time $t - \delta$ that is $\hat{y}' \equiv \hat{y}^{(t-\delta)}$
        Receive true label $y' \equiv y^{(t-\delta)} \in [0,1]$
        Incur loss $\ell_{\text{abs}} \leftarrow \left| \hat{y}' - y' \right|$
        Update weights $\omega^{(t+1)}_{i,r'} \leftarrow \omega^{(t)}_{i,r'} \exp^{-\eta_{r'} \cdot \left| \hat{y}'_i - y' \right|} \quad \forall_{i = \{i \in 1 \ldots N : \hat{y}'_i \text{exists}\}}$
        Update weights $\omega^{(t+1)}_{i,r'} \leftarrow \omega^{(t)}_{i,r'} \exp^{-\eta_{r'} \cdot \ell_{\text{abs}}} \quad \forall_{i = \{i \in 1 \ldots N : \hat{y}'_i \text{not exists}\}}$
```

3.5.3 Results

Several learners, also known as experts, are combined into a master model using the WMC variant described in Algorithm 1. It must be parameterized by the delay constant δ and the learning constant η_r, which is optionally dependent on the actual regime. Once the goal is to forecast 24 hours ahead, the real label is collected 24 hours after the prediction occurs. Therefore, the delay constant δ has a value of 48 half-hours. The learning constant η_r is set as one for every regime r. Additionally, regimes are defined annually as discontinuous and disjoint temporal intervals. When experts are combined, the weights learned are independent of the regimes. Table 3.9 summarizes the regimes used throughout the year. The algorithm starts learning (update the weights) from the (6 years long) training dataset and continues over the (2 years long) testing dataset. The accuracy of the algorithm is only based on the results of the last two years.

Table 3.9. Years are partitioned into several discontinuous and disjoint intervals called regimes. Each forecasting model might have better accuracy in a specific regime.

Regime	Description
R1	Christmas and New Year period (23 December to 2 January)
R2	Carnival period (Sunday to Wednesday including Tuesday Carnival)
R3	Easter period (Thursday to Monday including Easter Sunday)
R4	Other public holidays
R5	Weekends (except previous regimes)
R6	August (Summer holidays except weekends and public holidays)
R7	Other common days during Spring and Summer (except August)
R8	Other common days during Autumn and Winter

The experts are GAM-based models fitted with minor changes either by modifying the features of regression model or by fitting it against picked temporal periods, which comprise different training data. Equation 3.8 (GAMLF-SL-M2) is used as the first expert model A, and model

B is achieved by adding the 1 week lagged load. Finally, different training datasets are used to fit the same model structure, resulting in different GAM-based experts: models C-J. The training dataset is chosen according to the defined regimes and its temporal intervals. As shown in Table 3.10, it resulted in a set of predictors that perform well under specific periods or conditions. To evaluate the master model, the global RMSE and the global MAPE are also calculated in each iteration cycle after combining the newest predictor with the master model. The ensemble model I achieves better performance than the initial model A. The ensemble model's accuracy for one-day ahead forecast is about 154 MW for the global RMSE and 2.0% for the global MAPE.

Table 3.10. Both general-purpose and period-specific models were combined into a master model to minimize error (RMSE and MAPE). Units of RMSE are MW.

Model	**RMSE** (MW)	**MAPE** (%)	**Model introduced**
A	203.26	2.580	general-purpose model (Equation 3.8)
B	179.54	2.313	general-purpose model reviewed (including covariate that takes into account the demand 1 week ago)
C	169.74	2.182	weekends' model
D	163.91	2.094	August's model
E	165.03	2.083	public holidays' model
F	162.06	2.046	Spring and Summer's model
G	159.53	2.025	Easter's model
H	159.90	2.027	Carnival's model
I	**154.04**	**2.004**	Christmas and New Year's model
J	158.88	2.073	Autumn and Winter's model

The modified weighted majority algorithm (continuous version) proved to be an effective method to achieve better accuracy compared to a general-purpose predictor for STLF when GAM was used as a base technique. Although the predictors, which were combined, use the same regression technique – which is not required –, they have different specificity goals. Some predictors were specifically calibrated for special days or yearly/weekly repeatable periods and, therefore, they are more accurate on those specific days compared to general-purpose ones. Note that not only the regression technique was the same for all predictors, but also no new explanatory variables were added. Therefore, we conclude that the combination of the WMA while ensemble technique and the GAM while base modeling technique for predictors improves the overall accuracy. However, the WMC learning process includes no regulation, and thus overfitted predictors can adulterate the weights of the ensembling.

From the applicability perspective, though this method uses more data to train predictors, plus the initial ensemble training process, this is online learning. So, the method would decrease the importance of a specific predictor if its accuracy decays over time and vice versa. It is also important to remark that the ensemble layer provides an interpretable configuration: the weights impose the importance of each model within a regime, and the regimes are highly understandable by knowledge domain and acceptable by operators and managers. Therefore, this ensemble method does not decrease the degree of interpretability obtained from the basis predictors.

3.6 Conclusions

This chapter presents a methodology for the short-term load forecasting problem in the energy domain at the system level. The methodology consists of four stages: (i) formulation and selection of input variables as data preparation, (ii) definition of model structure, (iii) model calibration and tuning, and (iv) evaluation of model and residuals.

In the data preparation stage, exploratory data analysis and domain knowledge are used to explore the dataset and its characteristics, and formulate the input variables.

In the modeling stage, several models are presented, including GLMLF-B, GAMLF-SL, GBMLF-SL, and GAMLF-SLE. GLMLF-B is a benchmark model based on a generalized linear model. GAMLF-SL is an enhanced model built upon a generalized additive model, which incorporates additional lagged load and calendar components, in addition to the temperature components. GBMLF-SL is a gradient boosting machine load forecasting model. Lastly, GAMLF-SLE is an ensemble model that combines different learners based on the Weighted Majority Algorithm (WMA) and the GAM.

In the calibration and evaluation stage, error metrics and evaluation criteria are used to compare all models. The results indicate that GAMLF-SL outperforms the benchmark model, GLMLF-B, in terms of accuracy. Additionally, GAMLF-SL is an interpretable model that provides insights into the load forecasting problem. GBMLF-SL achieves the same accuracy as GAMLF-SL, but with two disadvantages: lower interpretability and a larger number of hyperparameters to optimize. GAMLF-SLE improves the overall accuracy of national load forecasting. The ensemble method presented in this chapter uses a collection of learners to enhance the accuracy of national load forecasting. The method combines the Weighted Majority Algorithm (WMA) as an ensemble technique and the Generalized Additive Model (GAM) as a base modeling technique for predictors. The modified weighted majority algorithm (continuous version) proves to be an effective method for achieving better accuracy compared to a general-purpose predictor for STLF when GAM is used as the base technique. Therefore, it can be concluded that the combination of the WMA as the ensemble technique and GAM as the base modeling technique improves the overall accuracy. From an applicability perspective, although this method uses more data to train predictors and involves an initial ensemble training process, it operates as online learning. Thus, the method reduces the importance of a specific predictor if its accuracy deteriorates over time and vice versa. It is important to note that the ensemble layer provides an interpretable configuration: the weights determine the importance of each model within a regime, and the regimes are highly understandable by domain experts and acceptable to operators and managers. Therefore, this ensemble method does not compromise the interpretability achieved by the base predictors. However, the WMC learning process does not include regularization, which means that overfitted predictors can affect the weights of the ensemble.

Chapter 4

Disaggregated Load Forecasting

This chapter describes the approach to individually predict the power load on thousands of assets following a disaggregated methodology. The object of the study is the power load in the secondary substations in Portugal. Approximately 30% are secondary substations that belong to a single consumer at high and medium voltage and 70% are secondary substations that feed dozens or hundreds of low-voltage consumers on a street or neighbor. The more disaggregated the forecasting is, the more complex and difficult it will be to predict. There is literature related to how to build forecasting models to a national or regional electricity consumption, but much less literature is available to low voltage load forecasting, other than smarter meters (LV consumers).

In particular, each secondary substation would have a trained model, although the structure of the model and the methodology are the same or very similar. Consequently, power time series for each asset are individually used to train the specific model.

4.1 Introduction

The disaggregated load denotes the power load in secondary substations, which is the interface between the medium-voltage (MV) and low-voltage (LV) power system. Smart meters that track this local power load are usually installed close to power transformers that have the function of stepping down a higher voltage to a lower voltage. This conceptually divides the grid into different voltage levels, and this chapter focuses on the forecasting problem of the “last mile” of the distribution network. Unlike primary substations that feed mainly a ring network scheme, in which more than one network path is possible through grid maneuvers to prevent outages when a primary feeder fails, secondary substations feed mainly a radial network scheme, which distributes the energy to the final consumers as a ”leaf“ section of the grid. It is also important to note that the secondary substation can be owned by a unique client, so the electricity measurements are from a specific customer’s site (point of energy delivery), usually a large building or an industrial customer. These secondary substations will be called PTC (client’s power transformer). Contrariwise, the secondary substations operated by the distribution system operator (DSO) are named PTD (distributor’s power transformer). For those, the electricity demand measurements refer to an aggregated set of consumers nearby, usually a neighborhood of a few dozens or even hundreds of low-voltage clients.

4.2 Data Overview

The data come from 96 989 secondary substations of Portugal mainland's distribution grid, the entire system. It consists of 26,479 client's (PTC) plus 70,510 DSO's (PTD) secondary substations. The latter feeds 99.6% of electricity consumers (points of energy delivery) in Portugal, which, however, represents 47.4% of the energy consumed on the mainland of Portugal.

Each secondary substation might have the whole or a subset of the six time series depicted in Table 4.1. The time series are 15 minutes resolution – that means 96 datapoints each day – and most is collected daily and made available centrally a few hours late. However, as with any distributed system in production, it may not be able to collect a few datapoints due to communications, maintenance, and other temporary issues.

Table 4.1. The time series collected by meters in secondary substations.

Field	Symbol	Description	Unit
1	A^+	positive active power	kW
2	R_i^+	positive inductive reactive power	kvar
3	R_c^+	positive capacitive reactive power	kvar
4	A^-	negative active power	kW
5	R_i^-	negative inductive reactive power	kvar
6	R_c^-	negative capacitive reactive power	kvar

Despite the fact that the DSO embraces several types of forecasting, including A^-, which represents energy generation from different generation technologies, for our purposes, this dataset focuses on positive active power A^+; although the other time series forecasts are equally important for grid planning and operation.

Numerical weather prediction (NWP) data is gathered with a 3 hour time step over a spatial grid box, updated twice a day (00 and 12 UTC) for the next 72 hours. However, the predictions are centrally available some hours late (between 7 and 8 hours after the base time to which the prediction refers). The geographical discrete points are arranged in a two dimensional regular grid located every 0.125 degrees over longitude and latitude, in which it covers whole Portugal mainland[1]. In this scenario, the challenge of assigning the most related weather station information to each secondary substation is minimized because there is a spatial grid box: Each secondary substation is associated with the closest euclidean-distant NWP, as shown in Figure 4.1. Table 4.2 shows the numerical weather variables collected that are available for forecast purposes.

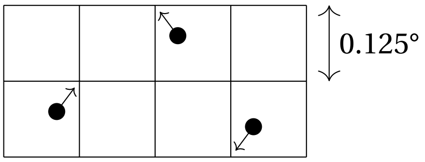


Figure 4.1. Taking into account the NWP over the spatial grid box, each of the three substations (circles) is associated with the closest Euclidean-distant NWP location.

Table 4.2. The numerical weather prediction collected for modeling and inferring purposes.

Field	Symbol	Description	Unit
151	msl	mean sea level pressure	Pa
167	2t	2 meter temperature	K
169	ssrd	surface solar radiation downwards (accumulated)[xiii]	$\mathrm{J\,m^{-2}}$
228	tp	total precipitation (accumulated)[xiv]	m
228246	100u	100 meter U wind component (towards east)	$\mathrm{m\,s^{-1}}$
228247	100v	100 meter V wind component (towards north)	$\mathrm{m\,s^{-1}}$

[xiii] SSRD is accumulated over a particular period since the initial time step. [xiv] Total precipitation is accumulated over a particular period since the initial time step. Units are the depth the water would have if it were spread evenly over the grid box.

Furthermore, due to the different time resolution between the original load observations (15 minutes) and the meteorological predictions (3 hours), the load observations are downscaled to 30 minutes using the average of its quarter-hour values, and the temperature is upscaled to 30 minutes using linear interpolation when the gap between two consecutive datapoints is not greater than 3 hours. Missing values are removed from the dataset.

4.3 Individual Secondary Substation Model – GAMLF-SSL

The GAM based load forecasting, GAMLF-SSL, is a regression model for secondary substations level. The approach aims to individually predict the power load on thousands of assets at a disaggregated level. The more disaggregated the forecasting, the more complex and difficult it is to predict. There is literature related to how to build forecasting models to national or regional electricity consumption, but much less literature is available for medium- and low-voltage forecasting (except for individual low-voltage clients). However, the same principles are applicable to the disaggregated STLF with the appropriate changes and further extensions.

4.3.1 Model

Given the dependent variable $\mathbf{y}_n$, the active power load (A$^+$) of the asset n in kilowatt (kW),

$$\mathbf{y}_n = \begin{bmatrix} y_{1,n} & y_{2,n} & \cdots & y_{t,n} & \cdots & y_{T,n} \end{bmatrix}^\top, \tag{4.1}$$

and the matrix $\mathbf{X}_n$ the calendar and meteorological variables that would be used as explanatory variables, where each entry $\mathbf{x}_{t,n}$ has a structure similar to that of Equation 3.6,

$$\mathbf{x}_{t,n} = \begin{bmatrix} x_t^{\text{(DayOfWeek)}} & x_t^{\text{(PublicHoliday)}} & x_t^{\text{(DayOfYear)}} & x_t^{\text{(TimeOfDay)}} & x_{t,n}^{\text{(Temperature)}} \end{bmatrix}^\top. \tag{4.2}$$

Besides the components already introduced in the previous Section, $x_{t,n}^{\text{(Temperature)}}$ is the last predicted temperature value for time t and the geographically closest point to asset n. Note that, although the actual temperature observation could be used during the fitting, that dataset was not available. On the other hand, during forecasting, the prediction of temperature is an input requirement.

[1] Latitude limits: $[36.5°, 44.0°]$; Longitude limits: $[-10.0°, -5.5°]$

The initial time t may differ for each asset n due to recent asset installations, but for most assets, the time t runs from January 2015 to December 2019. Individual datasets are partitioned into training and testing datasets so that the last year is used to calculate the accuracy metrics.

The structure of the model is based on Equation 3.9. The forecast of the power load at time t for the asset n is

$$\hat{y}_{t,n} = \begin{cases} \beta & |S_n| = 1 \\ \hat{y}^*_{t,n} & |S_n| > 10 \\ \hat{y}^*_{t,n} \underbrace{- f^{(\text{LagLoad24h})}\left(y_{t-48,n}\right) - f^{(\text{LagLoad1w})}\left(y_{t-336,n}\right)}_{\text{covariates removed}} & \text{otherwise}^2 \end{cases} \tag{4.3}$$

where $\hat{y}^*_{t,n}$ is the same equation 3.9 without the trend covariate,

$$\begin{aligned} \hat{y}^*_{t,n} &\overset{\text{def}}{=} \hat{y}^{(\text{M3})}_t - \beta_1 x^{(\text{Trend})}_t \\ &= \beta_0 + f^{(\text{LagLoad24h})}\left(y_{t-48}\right) + f^{(\text{LagLoad1w})}\left(y_{t-336}\right) \\ &+ f^{(\text{TimeOfDay})}\left(x^{(\text{TimeOfDay})}_t\right) \\ &+ \sum_{i \in G} \mathbf{1}_{\left(x^{(\text{DayType})}_t = i\right)} \left(\beta_i + f^{(\text{TimeOfDay/DayType})}_i \left(x^{(\text{TimeOfDay})}_t\right)\right) \\ &+ \sum_{j \in \{\text{NewYear},\cdots,\text{Christmas}\}} \mathbf{1}_{\left(x^{\text{PublicHoliday}}_t = j\right)} \beta_j \\ &+ f^{(\text{DayOfYear})}\left(x^{(\text{DayOfYear})}_t\right) \\ &+ f^{(\text{Temperature})}\left(x^{(\text{Temperature})}_t\right) \\ &+ f^{(\text{Temperature/TimeOfDay})}\left(x^{(\text{Temperature})}_t, x^{(\text{TimeOfDay})}_t\right) \\ &+ f^{(\text{Temperature/DayOfYear})}\left(x^{(\text{Temperature})}_t, x^{(\text{DayOfYear})}_t\right) \\ &+ \epsilon_t \end{aligned} \tag{4.4}$$

where $G = \{\text{MondayNoHoliday}, \ldots, \text{SundayNoHoliday}, \text{Holiday}, \text{HolidayOnWeekend}\}$.

Due to the capabilities of the energy meters, the measurements are actually non-negative integers, $\mathbf{y}_n \in \mathbb{N}_0^T$. However, this fact was not considered during the regressor modeling and during its accuracy assessment. There are assets that present a constant measurement $\mathbf{y}_n \in \{c\}^T$, or a small set of unique values, a small $|S_n| = \left|\left\{y_{t,n}\right\}_{t \in \{1,\cdots,T\}}\right|$. For those time series, a simpler regressor structure was used since the number of free parameters to estimate cannot be larger than the number of unique observations. Therefore, the third element of Equation 4.3 removes the covariates related to the lagged power load.

[2] Actually, for implementation purposes, this model is used when the original model "$\hat{y}^*_{t,n}$" returns an error due a small (undefined) $|S_n|$.

4.3.2 Results

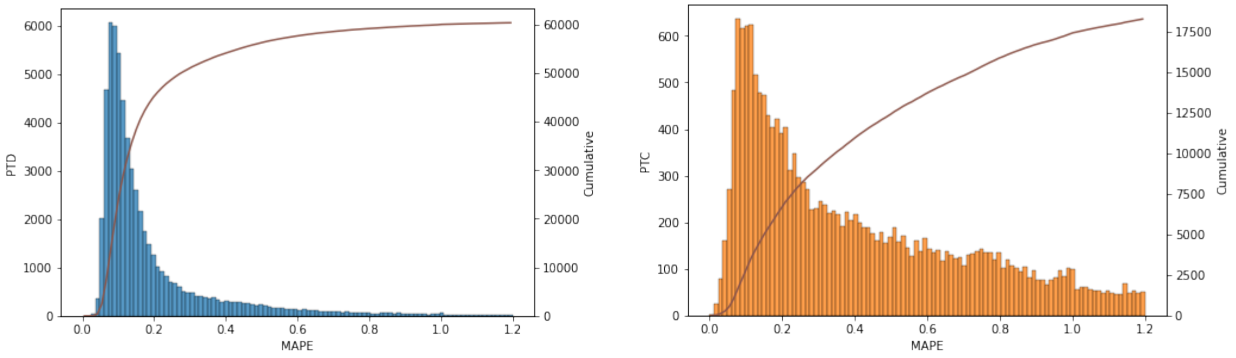
To assess the performance of $\hat{y}_{t,n}$, the generalized linear model-based load forecasting at the secondary substation level, the entire year 2019 was used as a test (see Figure 4.9 for an example of an asset). Note that one year is the minimum acceptable to test a forecasting model whose target value shows annual seasonality. There are several assets whose data was not available before 2019 for training and testing purposes. So, the following results consider the forecasting models for 22,974 PTC assets and 61,689 PTD assets.

Another relevant issue is the prediction horizon, which was set at 24 hours (48 half-hours). This means that for any time $t+48$ the model forecasts, the available explanatory variables refer to a time at or before t, except for the temperature predictions.

To accurately assess the performance of the forecasting models, several error metrics (MAE, MAPE, RMSE, NRMSE, R^2, MASE, APN, MAPN, NMAPN) were used for each asset forecast $\hat{\mathbf{y}}_n$. Only the scaled errors, MAPE, MASE, NRMSE, and NMAPN, were passably analyzed together and presented as histograms.

MAPE

The Figure 4.2 exhibits the MAPE histograms. Half of the models, which forecast the power load of the PTD, have a MAPE below 0.126, while only 16.35% of the PTC models present a MAPE below the same value. Both distributions are right-skewed, but the PTC error distribution tail is flatter than the PTD's. Therefore, the model structure is better designed to cope with the forecast of the PTD power load.



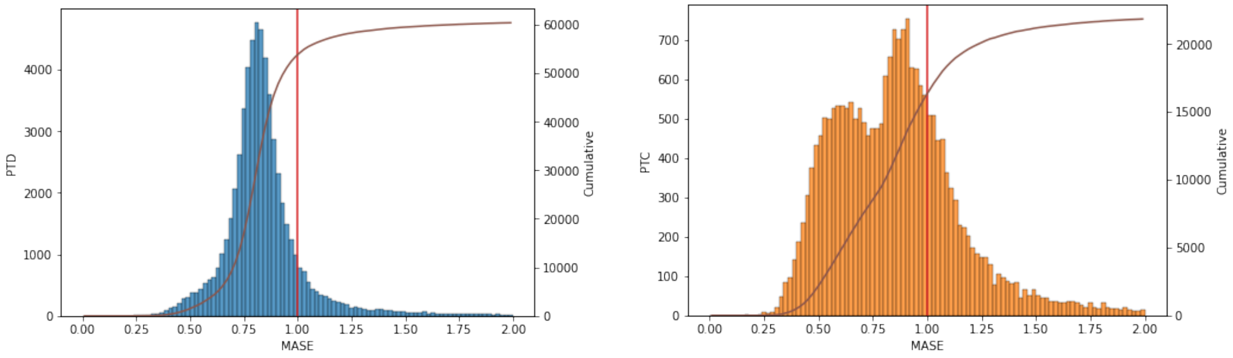
(a) PTD - MAPE

(b) PTC - MAPE

Figure 4.2. Mean Absolute Percentage Error for PTD and PTC models. Both distributions are right-skewed but with different flatness. The brown line indicates the cumulative count. The graph was cut in $e_{\text{MAPE}} = 1.2$, which means that 1366 PTD models (2.21%) and 4692 PTC models (20.42%) are out.

MASE

Figure 4.3 shows the MASE histograms. Note that MASE compares the model with a naive forecast method. Taking into account daily seasonality, the naive forecast method returns the power load observed 24 hours before, $\hat{y}_t^{\text{NAIVE}} = y_{t-48}$. The error metric is scale-free and suited to time series with zero or near zero values because it never gives infinite or undefined values except for the irrelevant case where $\forall_t \; y_t = C$. Both graphs also show a vertical line $e_{\text{MASE}} = 1$. Models are better than the naive model when $e_{\text{MASE}} < 1$. 82.8% PTD models and 66.0% PTC models are better than the naive model.



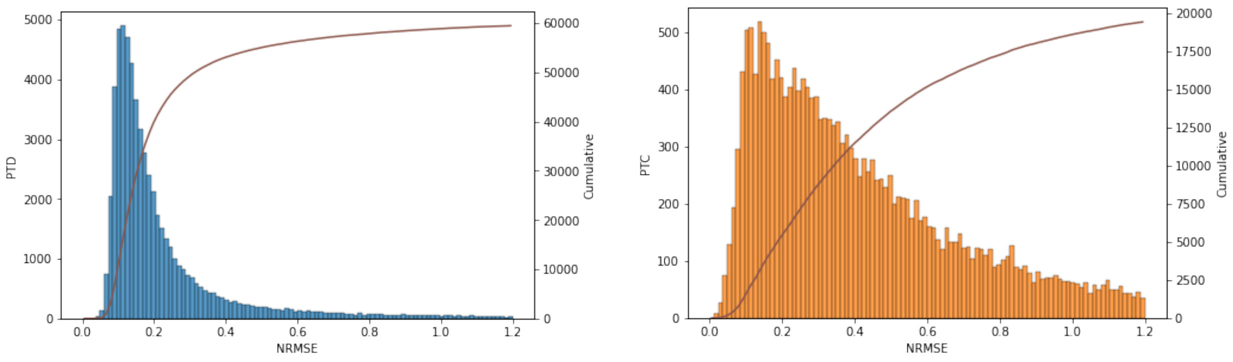
(a) PTD - MASE

(b) PTC - MASE

Figure 4.3. Mean Absolute Scaled Error for PTD and PTD models compared with naive forecast, which returns the power load observed 24 hours earlier. 53,391 PTD models (86.5%) and 16,083 PTC models (70.0%) are better than the naive model, $e_{\text{MASE}} < 1$. The brown line indicates the cumulative count. The graph was cut in $e_{\text{MASE}} = 2$, which means that 1,308 PTD models (2.21%) and 1,128 PTC models (4.91%) are out.

NRMSE and NMAPN

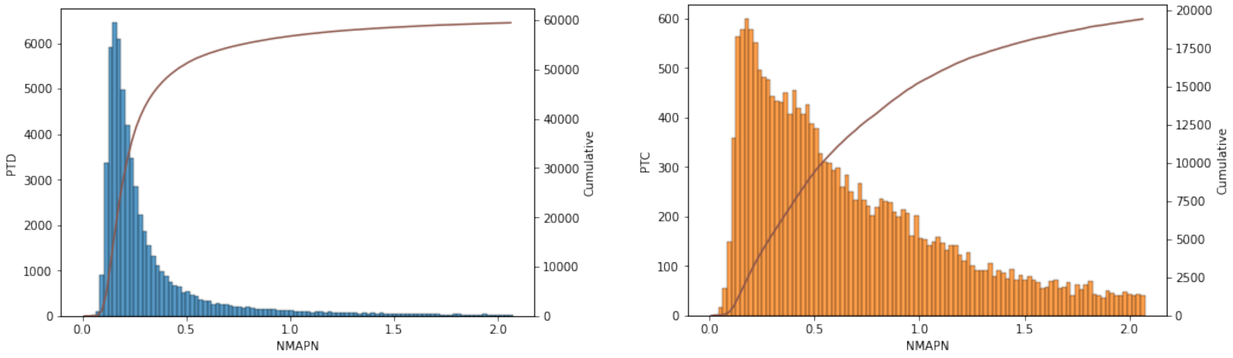
Figure 4.4 shows the NRMSE histograms and Figure 4.5 the NMAPN histograms. Half of the models that predict the PTD power load have a NRMSE below 0.162, while only 16.79% of the PTC models have an NRMSE below the same value. While, half of the models that forecast the PTD power load have a NMAPN below 0.222, while only 14.32% of the PTC models have an NMAPN below the same value. Both distributions show a long tail due to a percentage of assets that does not follow the general structure or the quality of time series is low (for instance, non-decimal values with a low amplitude).



(a) PTD - NRMSE

(b) PTC - NRMSE

Figure 4.4. Normalized Root Mean Square Error for the PTD and PTC models. Both distributions present a long tail. The brown line indicates the cumulative count. The graph was cut in $e_{\text{NRMSE}} = 1.2$, which means that 2,214 PTD models (3.59%) and 3,533 PTC models (15.38%) are out.



(a) PTD - NMAPN

(b) PTC - NMAPN

Figure 4.5. Normalized Mean Adjusted p-norm Error with $p = 4$ and $w = 3$ for the PTD and PTC models. Both distributions present a long tail. The brown line indicates the cumulative count. The graph was cut in $e_{\text{NMAPN}} = 2.0736$, which means that 2,257 PTD models (3.66%) and 3,580 PTC models (15.58%) are out.

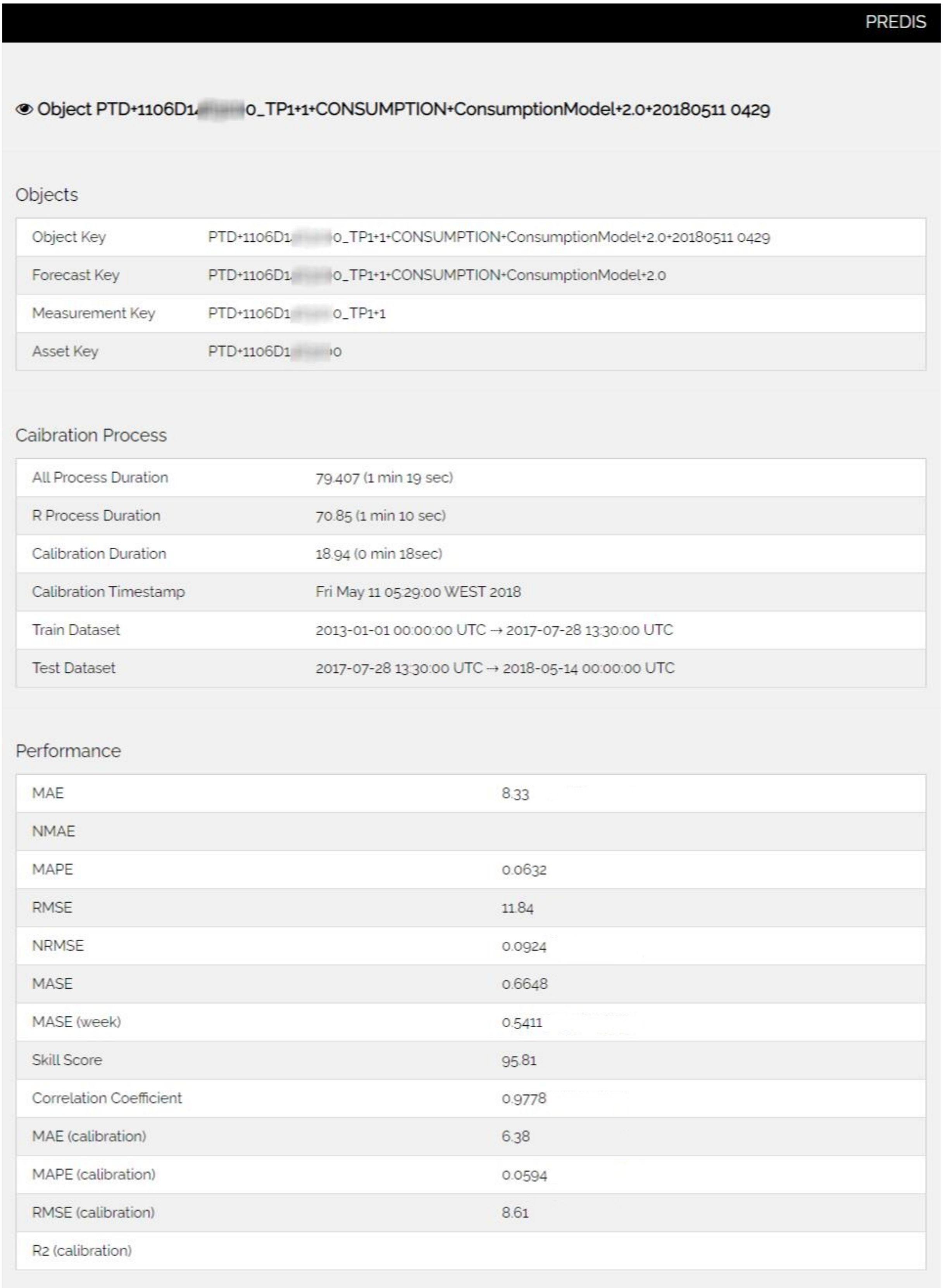
In general, the model is capable of forecasting such diverse amounts of time series (active power load) for all 84,663 secondary substations of the Portuguese mainland grid, in which data was available to train and test. It was concluded that this model is better than the naive model in 82.1% assets.

From an applicability perspective, fitting almost 100 000 models and daily forecasting takes a huge amount of computational resources. Chapter 7 introduces the distributed architecture of the daily forecasting system called PREDIS – PREvisão DIStribuída, capable of fitting the models in parallel computing and forecasting in a few two or three hours whose results arrived in useful time for the posterior operating processes. Furthermore, the fitting process of each asset results in a page with metrics and information: properties and accuracy metrics, load curves, residuals over time, residuals assumptions checking, accuracy across calendar, and best and worst week forecasting, as shown throughout Figures 4.6 to 4.10.

4.4 Conclusions

This chapter tackles a new data set that encompasses the 84 663 secondary substations of the Portugal mainland grid, where energy is converted from medium voltage to low voltage using power transformers. This dataset is already filtered by assets that do not have available data for the training and testing period. A GAM-based model is individually trained and evaluated considering accuracy, applicability, interpretability, and reproducibility.

The MASE metric recognizes the skill the model offers compared to the persistence model. Without removal of any asset or special days from testing dataset, 86.5% of PTD and 70.0% of PTC models are better than the persistence model. The PTC metrics reveal worse accuracy when compared with PTD's due to the fact that the latter feed dozens or hundreds of low-voltage consumers in a street or neighborhood whose aggregated consumption is more predictable than the single consumer of each PTC. The interpretability and reproducibility properties are inherited from the model and approach as described in the previous Chapter. The applicability is guaranteed by the distributed architecture of the daily forecasting system called PREDIS, which is capable in useful time to predict all individual time series.



👁 Object PTD+1106D1[redacted]0_TP1+1+CONSUMPTION+ConsumptionModel+2.0+20180511 0429

Objects

Object Key	PTD+1106D1[redacted]0_TP1+1+CONSUMPTION+ConsumptionModel+2.0+20180511 0429
Forecast Key	PTD+1106D1[redacted]0_TP1+1+CONSUMPTION+ConsumptionModel+2.0
Measurement Key	PTD+1106D1[redacted]0_TP1+1
Asset Key	PTD+1106D1[redacted]0

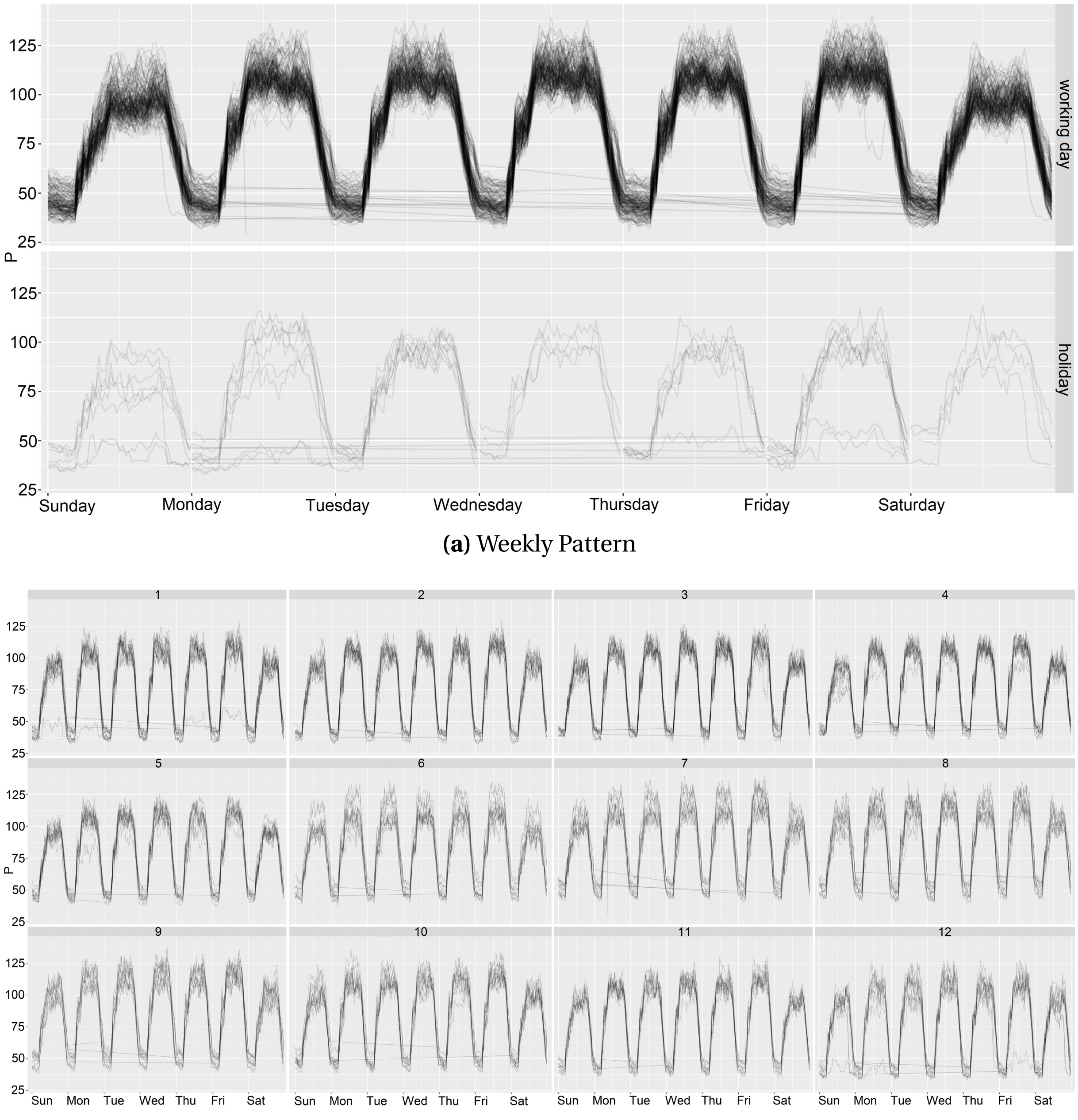
Caibration Process

All Process Duration	79.407 (1 min 19 sec)
R Process Duration	70.85 (1 min 10 sec)
Calibration Duration	18.94 (0 min 18sec)
Calibration Timestamp	Fri May 11 05:29:00 WEST 2018
Train Dataset	2013-01-01 00:00:00 UTC → 2017-07-28 13:30:00 UTC
Test Dataset	2017-07-28 13:30:00 UTC → 2018-05-14 00:00:00 UTC

Performance

MAE	8.33
NMAE	
MAPE	0.0632
RMSE	11.84
NRMSE	0.0924
MASE	0.6648
MASE (week)	0.5411
Skill Score	95.81
Correlation Coefficient	0.9778
MAE (calibration)	6.38
MAPE (calibration)	0.0594
RMSE (calibration)	8.61
R2 (calibration)	

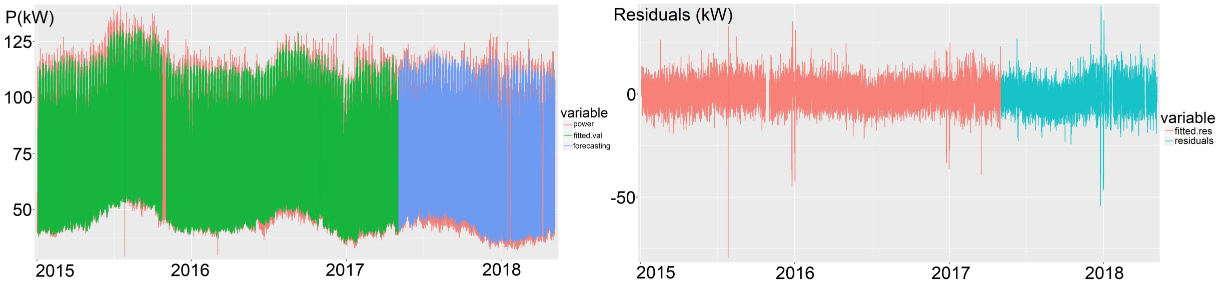
Figure 4.6. Each fitted model (artifact) has associated properties (primary and foreign keys, computing time, training and testing dataset intervals) and accuracy metrics to assess individually the artifact and its calibration process.



(a) Weekly Pattern

(b) Yearly Pattern

Figure 4.7. Hairball graphs allow you to get a perception in the daily, weekly, and yearly patterns and draw conclusions about why some patterns might not be captured by the model.



(a) Forecasting **(b)** Residuals

Figure 4.8. The load amplitude maintains across the year. At **(a)**, the actual values $\mathbf{y}_n$ are in red and split into two datasets (training and testing). The green and the blue are the predicted values $\hat{\mathbf{y}}_n$ in those two datasets. The residuals $\mathbf{y}_n - \hat{\mathbf{y}}_n$ are exhibited at **(b)** in which you can see peaks in a few spots.

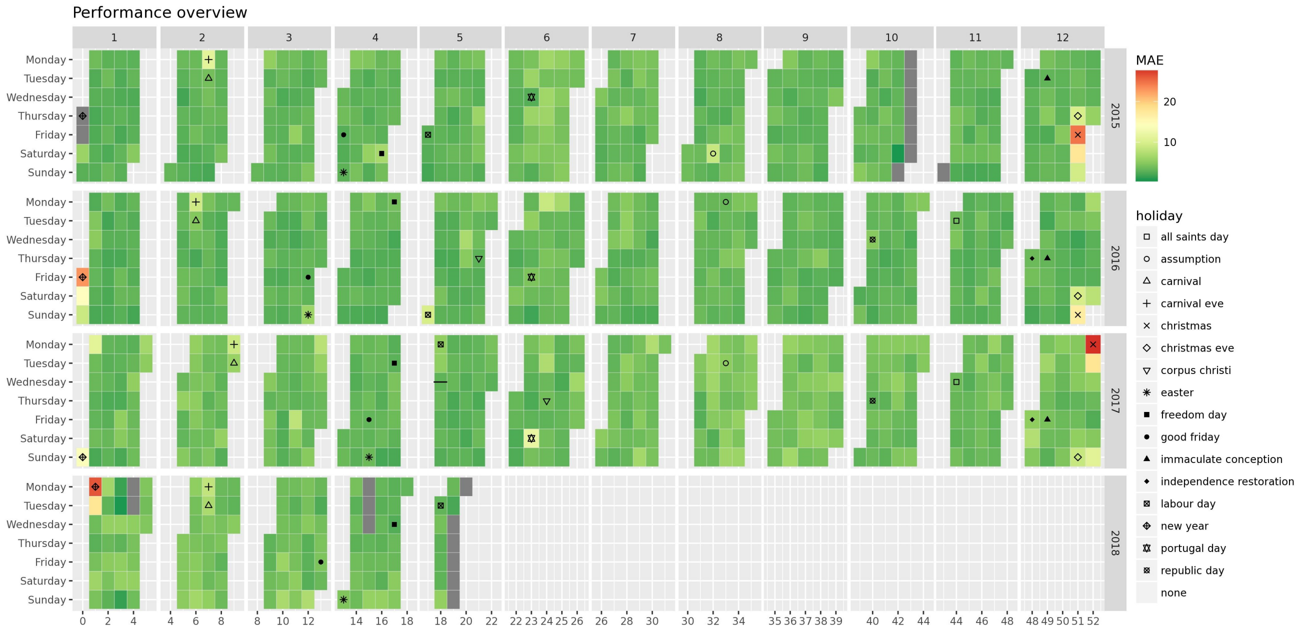
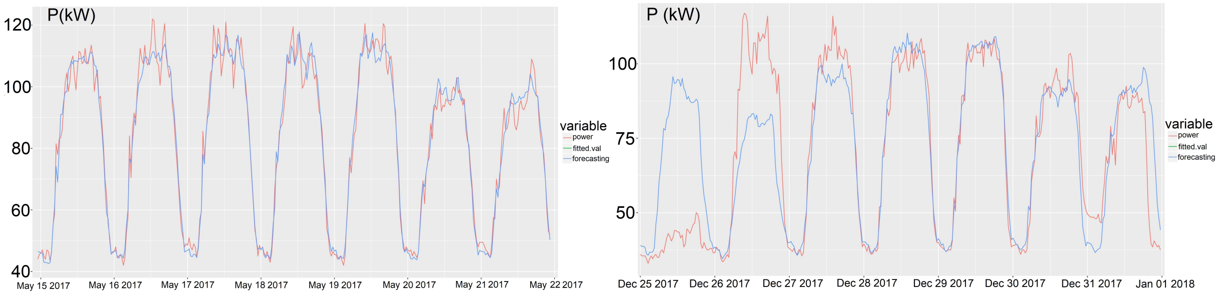
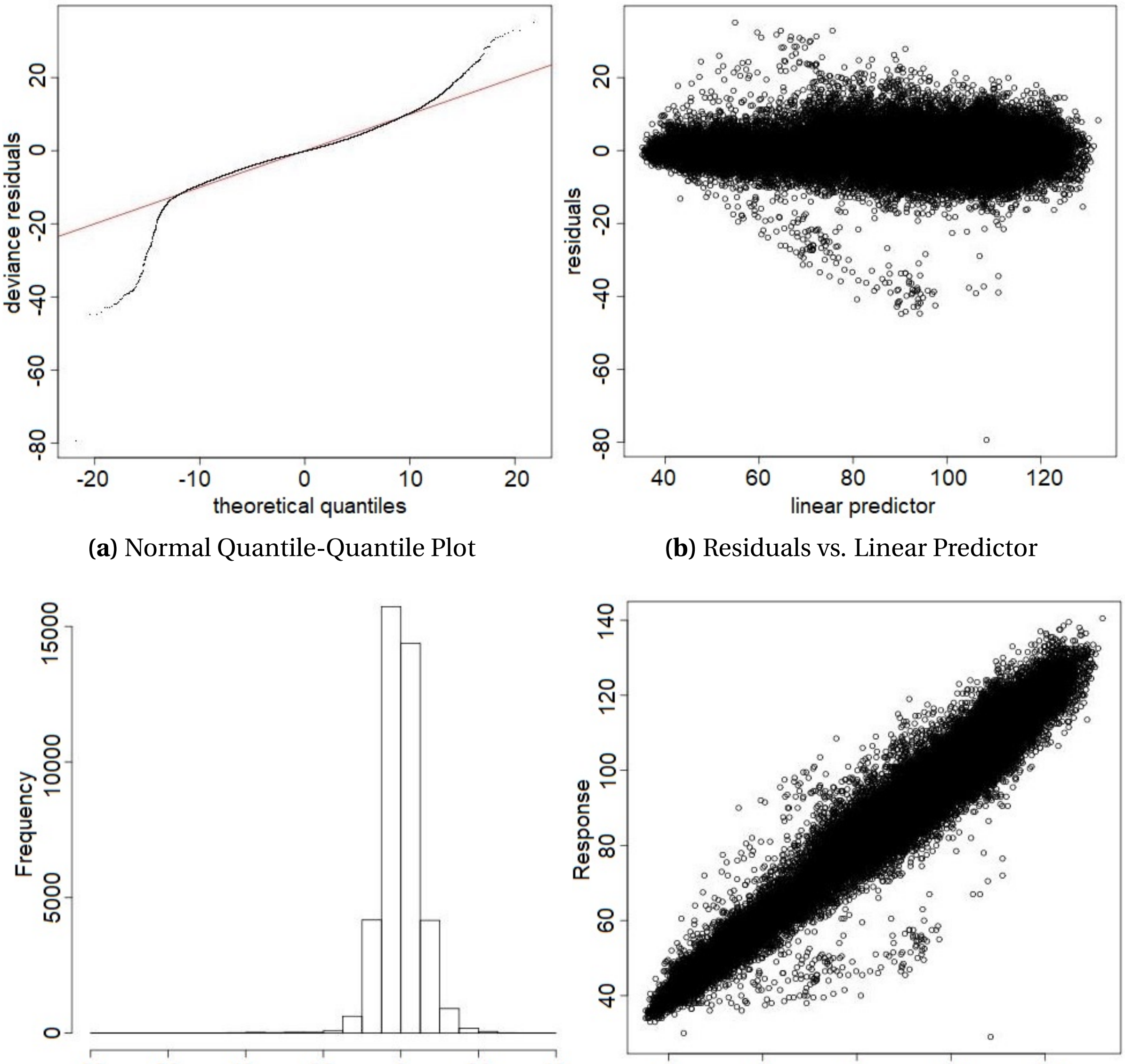


Figure 4.9. The performance calendar plot displays residuals throughout the year. Public holidays are displayed on the calendar with the respective symbols, as well as the time t that divides the training and testing datasets, marked by a horizontal line on 3 May 2017. Christmas and New Year, as well as the following day, show a higher error depending on the day of the week (weekend or not), which suggests that it is necessary to review the structure of the model in these combinatorial cases.



(a) Best forecasting week **(b)** Worst forecasting week

Figure 4.10. The best and worst forecasted weeks suggest patterns that the model did not capture.



(a) Normal Quantile-Quantile Plot

(b) Residuals vs. Linear Predictor

(c) Histogram of Residuals

(d) Response vs. Fitted Values

Figure 4.11. Graphs to check the plausibility of the assumptions. At **(a)**, the residuals $\mathbf{y}_n - \hat{\mathbf{y}}_n$ are sorted and the plotted against the quantiles of a standard normal distribution. Here, there are some specific residuals not following a theoretical normal distribution. The residuals $\mathbf{y}_n - \hat{\mathbf{y}}_n$, at **(b)**, and the predicted values $\hat{\mathbf{y}}_n$, at **(d)**, are plotted against the actual values $\mathbf{y}_n$. Though a constant variance in residuals is noticeable from the plots, there are residuals going outside the evenly scattered cloud around zero **(b)** or around the horizontal straight line **(d)**.

Chapter 5

Power Load Classifying using Shapelets

The advent of smart grids has increased power grid sensorization and so, too, the data availability at lower hierarchical power load levels. However, the more disaggregated the power load time series is, more complex and difficult is to forecast. It is important to consider shapes (patterns) presented in power load time series to cope with consumption diversity. This chapter considers the shapelet technique to create interpretable classifiers for four use cases at different hierarchical power levels (national, primary power substations, and secondary power substations). The use cases do not focus on the forecasting challenge, but on the ability to extract interpretable patterns and knowledge and embracing the interpretability of load classifiers.

5.1 Introduction

The electric power industry has been subject to constant changes. Utility companies have been aware of the threats and opportunities that arise as a result of this change. Climate action calls for the electric power industry to participate in the energy transition, once it is an important player in integrating more variable renewable energy sources and guaranteeing energy distribution for new needs, such as charging electrical vehicles, green hydrogen production, industrial heating electrification, and other opportunities for electrification all supported by renewable energy. The growing popularity of “behind the meter” on-site generation and storage, the new digital retail competitors that serve customers with bundle solutions towards energy-as-a-service business models, increased search for grid flexibility, and intense public and regulatory scrutiny are just some strengths that reinforce the need for more digitalization in the energy sector [90–93].

Data are an important asset that utilities have available today to support management decisions, excel in operational efficiency, and be more competitive. Moreover, data and technology to extract value from these data, generally addressed by artificial intelligence and machine learning, are no longer just a technological enabler, but rather an integrated part of the acceleration of the energy transition. The complex number of data sources along the value chain, led by higher levels of grid sensorization, has resulted in data streams whose value has not been fully explored.

Data-driven services within the energy sector pose challenges across regulatory, socioeconomic, and organizational (RSEO) aspects. Psara et al.’s review states that the value of data from various sources must be clearly understood in order to overcome identified organizational barriers related to the lack of data compatibility between different sources, the complexity of the data, and the inability to recognize the value of the data in addition to the siloed application in which data are collected [94].

Smarts grid deployment has led the industry sector to recognize the inherent value of data besides the billing and settlement functions: asset monitoring, behavior profiling, customer classification, load curve classification, and many other uses to be considered using the same dataset. The study of pattern and knowledge extraction might therefore help to understand human activities as energy consumers, raising the general understanding about energy demand, and thus helping not only today's power grid operations and decisions, but also define policies for tomorrow's operations [16]. By virtue of their function, smart grids gather data mostly in the form of time series, that is, observations of the same signal over time.

In the last decade, new series-based algorithms have been developed and new studies have analyzed the contribution of these algorithms to extract value and insights from series across various sectors. This includes sequential data such as sequences of numeric values, text, audio, and even image. If the sequence is time-stamped, the sequence is generally named a time series. Time series forecasting is an major area with extensive literature that infers or estimates further steps in the series [95]. The clustering of time series seeks to discover temporal patterns with an unsupervised approach [96]. On the other hand, with a supervised approach, time series classification considers discriminatory features dependent on the ordering to organize time series into predefined labels or classes [97]. A set of data mining techniques for time series has also been developed: symbolic representation [98], motifs [99], discords [99], shapelets [99], time series chains [100], snippets [101], semantic segmentation [102] and so on.

When considering the energy-specific domain in the application of those techniques, shapelets is one that was not extensively applied. From the literature review, shapelets were used for non-intrusive load monitoring (NILM) [103], discovering customer weekend load patterns [104], classification of district heating substations [105], evaluation of voltage stability [106–108], and clustering power curves [109] with a modified version of shapelets to work as an unsupervised technique. Although with a few applications, it is evident that shapelets have not yet extensively assessed time series with a power load at the national level, as well as at primary and secondary substations. This chapter aims to fill the gap in the literature on the energy domain, exploring the shapelet technique with four different goals with the same dataset. The dataset is related to the power load measured quarter-hourly at three levels: national, primary, and secondary substations. The data and the method are fully described in Section 5.3. Section 5.5 provides four use cases: (i) which pattern identifies weekend load curves from business days, (ii) which load pattern identifies Mondays from the rest of the business days, (iii) a classifier capable of identifying the load dynamics due to maneuvers across the grid, and (iv) a classifier capable of identifying the type of energy consumption from just the daily load curve. The study assesses whether shapelets are a technique capable of responding to the four challenges, and thus reinforces the value of smart grid data in addition to the siloed application in which data are gathered.

The most important added benefit provided by this study is the demonstration of the value and information that can be extracted as interpretable patterns from one of the most meaningful types of data collected in the energy sector, the load time series. Using appropriate machine learning techniques—in this case, shapelets—it is possible to extract value, which reinforces the importance of multi-source data-driven services within the energy sector and across its value chain. Section 5.6 draws conclusions and discusses the results.

5.2 Related Work

Time series do not have explicit features. For instance, each data point is a value, but the pattern or trend depends on the data points close to it. Whereas most classifiers consider discriminatory features non-dependent on the ordering values, which can be poisoned by even low levels of noise and distortions, the shapelets are local features, and thus they capture patterns as subsequences inside a time series able to discern different classes.

Shapelets were introduced by [110] as a new primitive representation of a time series that is highly representative of a class. At that time, they introduced shapelets as interpretable, more accurate, and significantly faster than state-of-the-art classifiers. Instead of looking for the global shape of time series and calculating the distance between them and a class representative, the shapelets technique only compares a local subsection of the shape that is particularly class discriminating. Shapelets have been continuously improved. The enormous quantity of shapelet candidates makes brute-force (exhaustive) shapelet discovery very slow for large datasets. Early abandon of Euclidean distance calculation combined with early entropy pruning were ideas introduced in the original paper [110]. Additional articles emphasize the reuse of computations and the pruning of the search space [111], while the projection of time series to the SAX representation was also elaborated by [112]. Furthermore, [113] proposed a novel fast method that avoids measuring the prediction accuracy of similar candidates through an online clustering/pruning technique. Although the shapelets concept was originally considered a supervised technique, it was exploited and extended that shapelets can be learned from unlabeled time series and used for unsupervised clustering [109]. A recent study proposes dynamic shapelets in which the differing representative of a long time series is considered in different time slices, as well as the evolution pattern of shapelets [114]. The evolution of shapelets is modeled as a graph, which represents how a time series evolves in terms of shape and interpretable patterns throughout time. To validate, the authors conducted experiments based on five time series datasets from different domains.

Shapelets were used in a few use cases in the energy domain with relative success. Recognizing the potential of shapelets as an interpretable technique for studying power load curves, the literature is not as extensive as one would like. Non-intrusive load monitoring discerns the individual electrical appliances of a residential or commercial building by disaggregating the accumulated energy consumption data without improving the individual sensorization of each appliance. The introduction of shapelets to this problem was proposed by [103], which starts with shapelets discovering from the current signatures of the recorded device instances present in a labeled database. The 60 Hz recorded samples consider the first few seconds of a device operation to search the shapelet instead of the entire time frame. Classification of district heating substations using shapelets as feature extraction was introduced by [105]. In this study, the classifier models are improved after the use of augment features extraction from the shapelet transformation. The transformation is not described in detail, but generally consists of the difference between the original time series and the shapelets previously discovered from them. Another author has applied the shapelets technique in the assessment of voltage stability [106–108]. With the development of smart grids, phasor measurement units are massively available, and so are the data, making it possible to use data mining techniques such as shapelets. In this case, the author has to deal with classes that are imbalanced and incrementally updated, and with the online application of the model. Another study, which approximates to ours, analyzes the effectiveness of the shapelet algorithm in classifying various weekend consumption patterns extracted from real-life data [104]. The author suggests the potential of shapelets to determine which customers use more electricity at weekends, a period in which consumption usually reduces, which saves during peak hours, or who responds to demand response events. However, the assessment considered only the weekend pattern discovery using 15 minutes of the consumption data of 33 buildings with different noise levels. None of the studies applied the technique to power load time series at different levels of aggregation (national, primary, and secondary substations) considering a systematic method and evaluation.

5.3 Data Overview

Three datasets of power load were used to perform the study of time series classification using shapelets. Dataset I and Dataset II are the same as described in Sections 3.1 and 4.2, though with a higher resolution (15 minutes). Unlike Dataset II, Dataset III contains measurements from assets in a higher hierarchical level of power system; it denotes the primary substations power load over 3 years.

Due to the dynamics of the distribution grid and the possibility of shifting the power load from a primary substation to another according to the management of the grid operator, there exist, particularly in dataset III, high load peaks and periods where the substation is shut down and the measure values are zero, as shown in Figure 5.1.

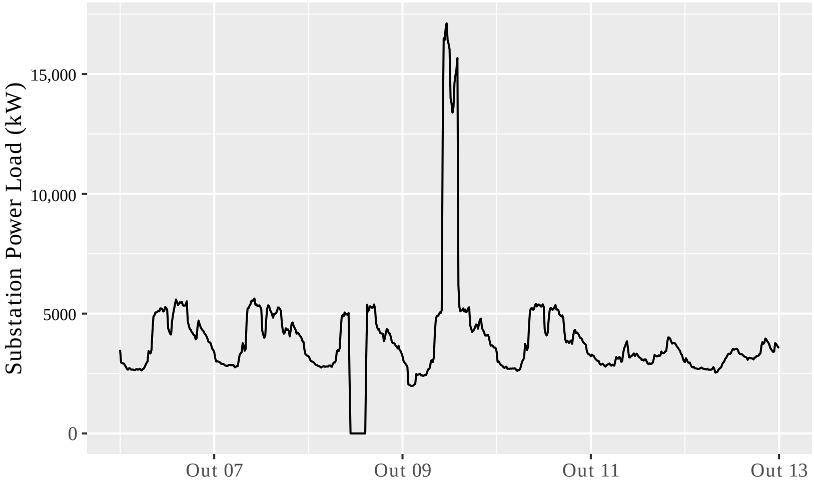


Figure 5.1. The substation load from Monday to Sunday with a maintenance shut-down on day 8 and a high load peak due to a load shift maneuver on day 9.

At the secondary substation level, annual, weekly, and daily seasonalities are generally kept, as at national level. However, as the secondary substations feed a high-demand building, venue or industry, or even a residential neighborhood, the load curve is more unpredictable when compared with higher levels (primary power substations and national levels) and patterns might be found in accordance with the specific usage of electricity at that point of energy delivery. Generally, unpredictability and noise increases as we move down the level hierarchy (national, primary and secondary substations).

Additionally, a calendar dataset was available with the following information: (i) day of the week, (ii) public holidays of Portugal, (iii) local statutory holidays for Portugal's municipalities, (iv) strike days, and its local disrupted services when applicable.

5.4 Method

Four use cases are set following a systematic method. First, a scenario is set that includes the question to be answered. The question is usually related to whether an interpretable pattern is discoverable by shapelets, which explains how two or more classes are discriminated. Second, the appropriate dataset is chosen with the respective data wrangling to set up the time series and its labels. Usually, the time series are transformed into daily load curves and chosen to balance the classes. For example, for the weekends use case, the daily national load curves are randomly selected within each class (weekend and business day). Daily curves that present outliers or missing data are kept out of the training and testing datasets. Third, the method applies the steps of the shapelet algorithm: (i) a pool of candidates is created from the time series inputs and minimum and maximum shapelet-length parameters, (ii) the best performing candidates are ranked using the information gain criterion over the target (or other prediction quality metrics like the Kruskal–Wallis or Mood's median [115], or F-Stats [116]), (iii) the chosen best performing candidate is used as a tree node to create a tree-based model interactively, (iv) the time series input are split using the model built so far and a new iteration starts on each of the new leaves, (v) the algorithm stops when pruning parameters are met or a whole leaf time series corresponds to a class. Finally, it is checked if the shapelets were able to answer the initial question, and the performance of the resulting decision tree classifier is evaluated against the testing dataset through the accuracy metric, $\frac{\sum_k TP_k}{N}$, where TP_k are the objects correctly classified by the true label k, and N is the number of objects.

Definitions

A simple symbol sequence is an ordered list of symbols of a given alphabet. The dataset used in the use cases is a sequence of real values. So, we are actually interested in classifying time series. Time series can be univariate or multivariate. We considered only the simplest.

Definition 1. A **time series** T is a sequence of real values typically ordered in ascending order by timestamp. For example, $T = \langle (t_1, v_1), (t_2, v_2), \ldots, (t_l, v_l) \rangle$ is a simple time series of length l that records data points from time t_1 to t_l. A **subsequence** S of time series T is a sampling length $m \le l$ of contiguous positions from T, that is, $S = \langle (t_p, v_p), \ldots, (t_{p+m-1}, v_{p+m-1}) \rangle$ for $1 \le p \le p + m - 1 \le l$.

Definition 2. The **distance from a subsequence to a time series**, $\mathscr{D}(S, T)$, returns a non-negative value which is the Euclidean distance (or other distance function) between S and its best matching location somewhere in T, that is, where the distance is minimum.

Definition 3. A **dataset** D of length n is a set of time series T_i and its class label c_i. Formally, $D = \langle T_1, c_1 \rangle, \ldots, \langle T_n, c_n \rangle$ and $c_1, \ldots, c_n \in C$, the set of possible labels. $\#C_i$ is the number of time series in class C_i.

Definition 4. The **entropy** $\mathscr{E}$ of the dataset D is defined as $\mathscr{E}(D) = -\sum_{i=1}^{\#C} p_i \log_2(p_i)$ where $p_i = \frac{\#C_i}{n}$ is the probability of the class.

Definition 5. A **split** is a tuple $\langle S,\theta\rangle$ of a subsequence S and a distance threshold θ (or separation gap) that separates the dataset into two smaller datasets, D_L and D_R. When a split separates the dataset with the maximum information gain, the subsequence S is denominated as a **shapelet**.

Definition 6. The **information gain** $\mathscr{G}$ of a split strategy that divides D into two subsets D_L and D_R of length n_L and n_R is given by $\mathscr{G} = \mathscr{E}(D) - \frac{n_L}{n}\mathscr{E}(D_L) - \frac{n_R}{n}\mathscr{E}(D_R)$.

Classifying with a shapelet S and its corresponding separation gap θ produces a binary decision node on whether a time series belongs to a certain class or not. The shapelets are embodied on the nodes of a decision tree to create a universal classifier. At each step of decision tree induction, the shapelet and the corresponding split gap are computed on the training subset considered in that step [110].

5.5 Results

This section will describe four use cases based on energy load time series of different power levels. Four particular classification problems were studied in order to assess decision trees using shapelets applied to energy data.

5.5.1 Weekends

The first classification problem is related to the lower consumption that occurs on weekends. In this problem, the goal is to discover whether a 24 h time series corresponds to a weekend or a business day. As described above, the national load follows a very distinctive trend over the weekend, but even these are different over the year.

The input data are a set of time series of 96 points each (24 h with a quarter-hour resolution), extracted from the 8 year national load. Each time series has a binary classification that identifies whether it is from a business day (label A) or a weekend (label B)—Table 5.1. Public holidays were separated.

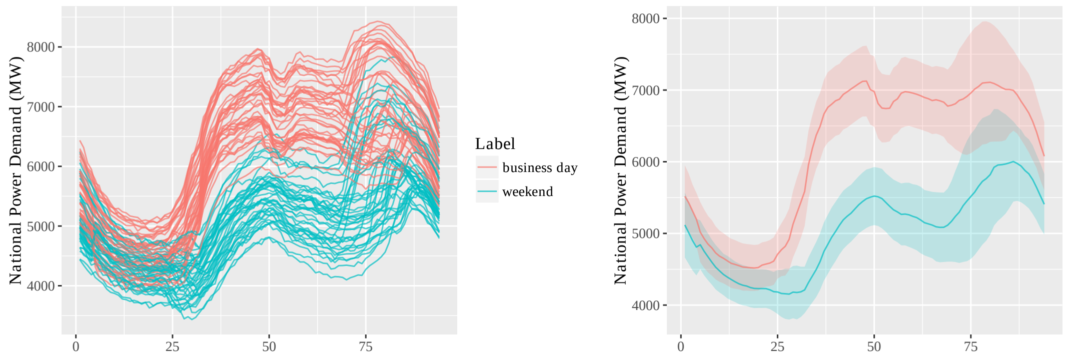
Table 5.1. The number of time series in the datasets for a business day (label A) and a weekend (label B).

Dataset	Label A	Label B	Total
Train	40	40	80
Test	2014	772	2786

Figure 5.2 shows us the training dataset. Note that demand throughout the daylight period is smaller on weekends, and the morning ascending occurs later and demand peaks appear at different times in the day. On the other hand, on business days, there is a demand break around lunch. We ask whether the discovery algorithm will find the best shapelet(s) to distinguish those time series.

For this scenario, the shapelet discovery algorithm is parameterized with a minimum and maximum shapelet length of 20 and 35 positions. This corresponds to shapelets between 5 and 8.75 h long, a length that we have assumed to be enough to find the shape of weekend consumption.

Figure 5.3 shows the decision tree obtained after the training phase. One shapelet proved to be enough to decide whether a time series T is from a weekend or not. This results in an

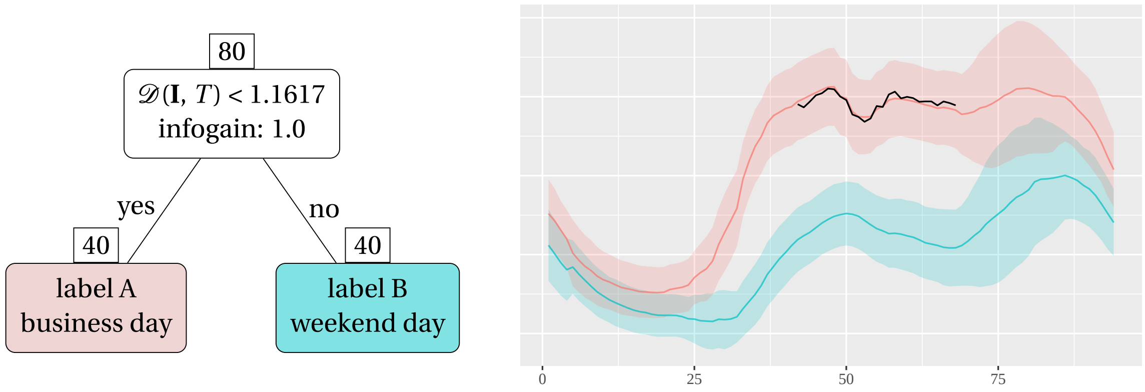


(a) training dataset

(b) mean and standard deviation

Figure 5.2. The complete training dataset with 40 time series of each class.

information gain of value 1.0 in the root node. Thus, during classification, all normalized subsequences of an arbitrary time series T are compared with the shapelet I. If just one normalized subsequence exists whose distance to the shapelet is less than 1.1617, then the time series is classified as label A, a business day.



(a) decision tree

(b) shapelet I

Figure 5.3. The decision tree classifier for the weekend problem. During the classification of an arbitrary time series T, its normalized subsequences are compared to the shapelet I (the black line). It is classified as label A if one of its normalized subsequences is *similar* to the shapelet. Note that the shapelet I was positioned according to where it was found, just to match the *similarities*. The shapelet might have a different scale and does not correspond to the coordinate axis.

The resulting shapelet is 27 points long, and it is positioned in the middle of the time series, i.e., in the middle of the day. As one can see, the shapelet has grasped the demand break towards lunch—a trend which does not exist in the weekend time series. Note also that amplitude differences among the consumption time series, due to the effect of temperature or other human behavior, have not affected the accuracy of the decision tree.

Finally, the decision tree was evaluated with the testing dataset and it was found to perform with an accuracy of 96.77%.

5.5.2 Early Monday Morning

The aim of the second classification problem is to determine whether the following result can be found using the shapelets technique. There are several publications about short-term energy forecasting that use the day of the week as an explanatory variable following this idea: Mondays are different from the rest of the business days. They detach Monday into an individual category followed by the weekend category and the rest of the days of week as another category. The rationale behind this approach is the existence of differences between the consumption during Monday's first hours when compared with the other days, because Monday dawn comes from the end of the weekend which presents a different curve pattern as seen in the previous section. Is the shapelet algorithm able to discern the pattern presented in the Monday load curve from other business days and weekends?

The input data are similar to the previous scenario—time series of national demand and 24 hours long—but, this time, the target variable has three labels: Monday (label A), weekend (label B), and other business days (label C)—Table 5.2. Public holidays were separated from the dataset.

Table 5.2. The number of time series on the datasets from the category Monday (label A), weekends (label B), and other business days (label C).

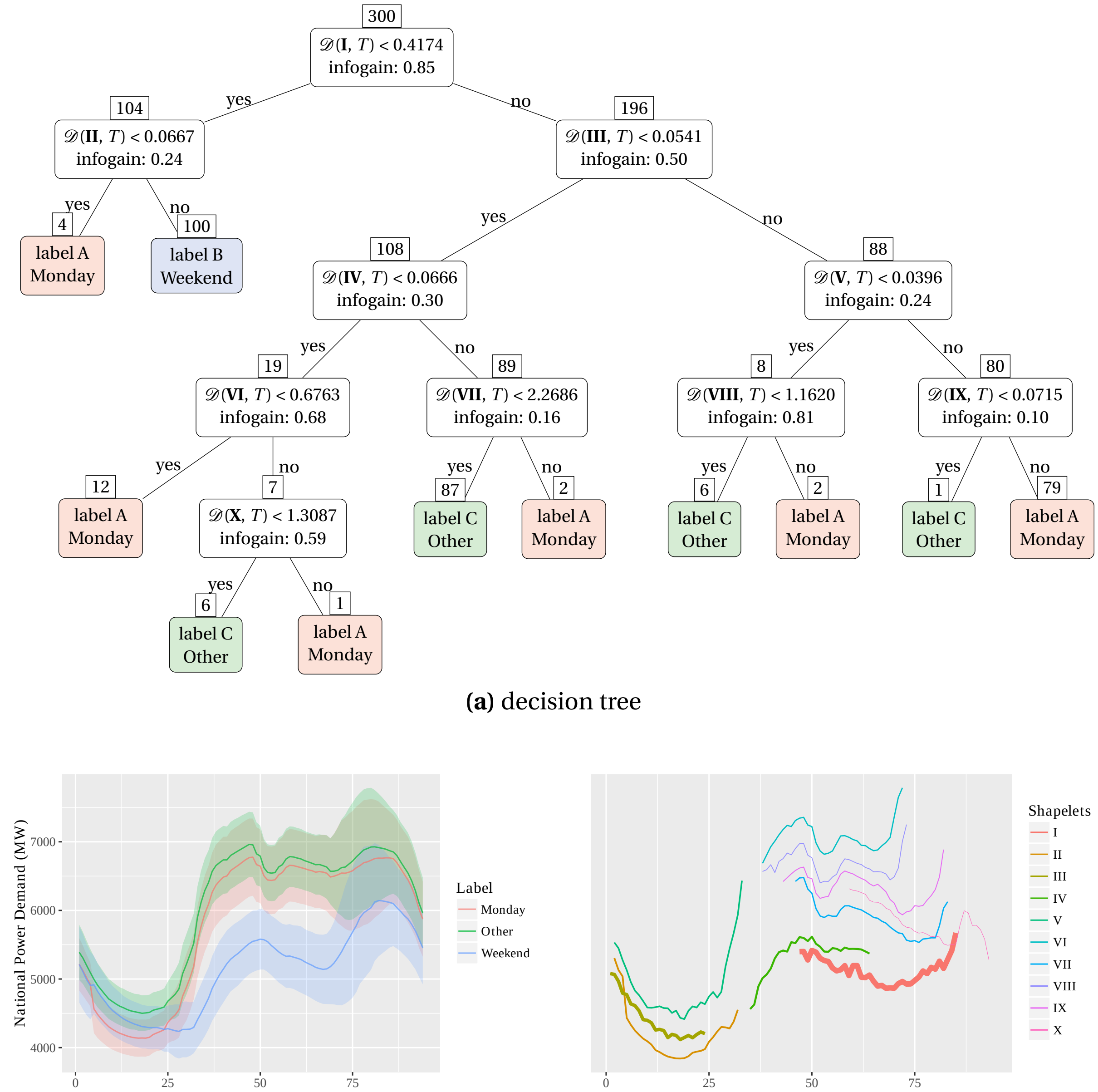
Dataset	Label A	Label B	Label C	Total
Train	100	100	100	300
Test	309	712	1545	2566

From the training dataset—Figure 5.4b—the weekend consumption pattern is distinguishable from the other days, as we saw in the previous section. However, the Monday pattern is very similar, except during the early morning. Monday's mean demand follows a different pattern when compared with the other business days: the very first hour follows the same pattern, but the time series dives further until the first daylight hours.

The resulting non-pruned decision tree—Figure 5.4a—has two important nodes that roughly split the training dataset into the three labels: the ones that use shapelets I and III. Actually, these are the nodes with the highest information gain multiplied by the number of time series to split (the number that appears above the node). In Figure 5.4c, the weight of the shapelet's line reflects its importance. Therefore, shapelet I splits the weekend time series from the dataset (the left most branch of the decision tree). Shapelets III and V, which occur in the early morning, capture a pattern that favors class C (the opposite side of the most right branch of the decision tree). Meanwhile, the shapelet II, which also occurs in the early morning, makes a final split between the weekend time series and the Monday time series.

The non-pruned decision tree uses other similar shapelets (from VI to X), but they are not easily interpretable. However, even if those shapelets were ignored, i.e., the decision tree was pruned, the model would get right 278 times of 300 items (94%) from the training dataset.

Finally, the testing dataset was used to assess the accuracy of this decision tree. Despite its unpruned nodes and potential overfitting, the decision tree has a high accuracy of 88.74%. Evidently, there is a rationale to split Monday from the rest of business days when forecasting energy time series: the Monday demand is different from the other days and the shapelets discovery algorithm was able to find the pattern.



(a) decision tree

(b) mean and standard deviation of training dataset

(c) shapelets

Figure 5.4. The decision tree classifier for the early morning pattern on Monday morning. (**a**) represents the non-pruned decision tree to classify time series as Monday (label A), weekend (label B) or other business day (label C) using the shapelets exhibited in (**c**). The line weight of the shapelets is proportional to its importance, i.e., the information gain times the number of time series splits in each node. (**b**) shows the mean and standard deviation of the training dataset with 100 time series of each class.

Furthermore, there are publications that break the business day, not into two categories but into three categories: (i) Monday, (ii) Friday, and (iii) Tuesday, Wednesday, and Thursday. We have tested this scenario, but the algorithm was not able to find a good shapelet that could accurately split time series. Indeed, the Friday time series shape is similar to Tuesday's, Wednesday's, and Thursday's. In that scenario, the same method resulted in a low accuracy of 58%.

5.5.3 Load Dynamics in Substations

Power substations have the function of transforming energy from very high voltage (VHV) to high voltage (HV), or from high voltage to medium voltage (MV) and each usually feeds thousands of clients. As a redundant system, the power grid can manage the shutdown of a particular substation, whether due to scheduled maintenance or failure. In this situation, the load of a substation is shifted to another and thus holds out the energy flow without any significant outage. Obviously, the load time series reflects these maneuvers. While one substation reflects a very high load peak, the observations of the other reveal a stationary no-load state, see Figure 5.1.

Rebuilding the historical state of this dynamic grid at a particular moment in the past can be a large computational problem due to the enormous amount of assets that change states. However, this information is important to energy forecasting tasks because of the high impact on loads due to load shifts.

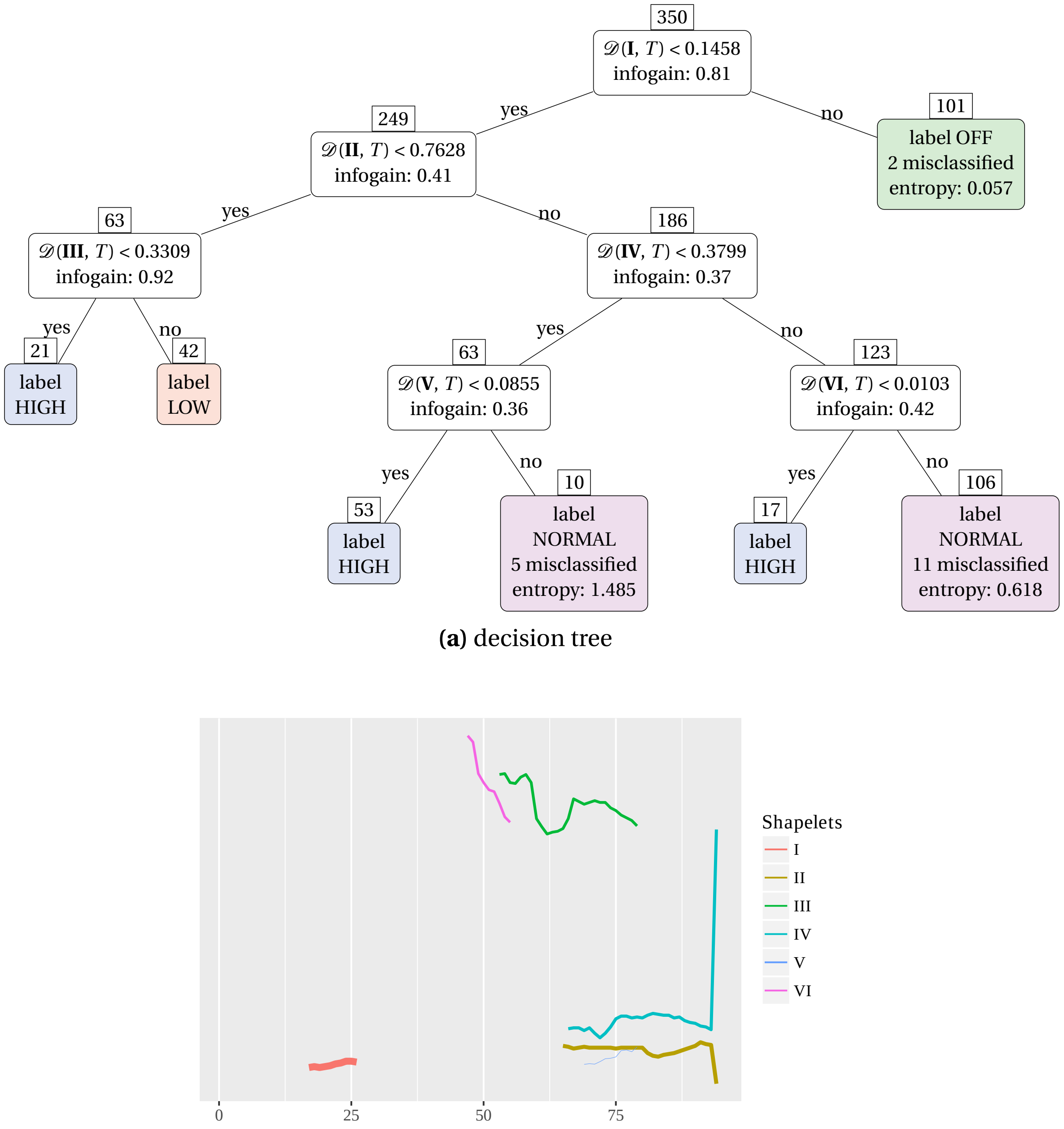
In this third classification problem, we have the goal of classifying whether a substation is in one of four states looking only for the load series. Input data are a set of time series extracted from three years of historical data through a 96-point sliding window (24 h). Each time series has a label that identifies the state of the substation at the 96th point—Table 5.3.

Table 5.3. The number of time series in the datasets for the states OFF, LOW, NORMAL, and HIGH. The first dataset has an approximate distribution along the labels when comparing the training and testing dataset. The second includes all available time series.

Dataset		Label OFF	Label LOW	Label NORMAL	Label HIGH	Total
1	Train	100	50	100	100	350
	Test	400	37	400	400	1237
2	Train	150	50	150	150	500
	Test	7893	37	121 139	787	129 856

Figure 5.5 shows the decision tree obtained after the training phase using the first dataset. Shapelets II and IV discovered the load shifts that occurred when the grid system maneuvers the substation.

The decision tree used on the testing dataset was the non-pruned. In conclusion, the decision tree calculated with the first dataset has a high accuracy of 90%, and 83% for the decision tree using the second dataset. Furthermore, we have made experiments using sliding windows of 9 points long, SAX preprocessing of time series, fixing a state switch three points before the end of the series, or modifying the minimum and maximum lengths of candidate shapelets. However, the accuracy of the decision trees was lower, between 48% and 77%.



(a) decision tree

(b) shapelets

Figure 5.5. The decision tree classifies the state of specific substations. (**a**) represents the classifier to reason the state of specific substations among four possible classes—OFF, LOW, NORMAL, or HIGH—using the shapelets exhibited in (**b**).

5.5.4 Type of Power Consumption

The curve of power consumption over a day depends on the purpose for which the energy is consumed. In this fourth classification problem, our objective is to classify whether power consumption serves one of these five purposes: (a) household, (b) industry, (c) services, (d) utilities, and (e) transportation. The demand curves are collected by meters of secondary substations owned by a distribution company or a high-demand client. Is the shapelets algorithm able to build a classifier capable of identifying the type of energy consumption from just the daily load curves?

The consumption data therefore refers to several sets of households or individual high-demand clients, for example: hospitals, retails, banks, post offices, government services, train and subway systems, waste water treatment plants, water pumping, gas utilities, and glass, furniture, rubber, plastics, porcelain, textiles, coating, and other types of industry. Table 5.4 shows us the number of meters available by class and the number of time series extracted from them. Note that the training data were obtained from meters other than the ones used for testing. Therefore, the problem is challenging and difficult to classify, but this may be due to the low number of meters available. Additionally, the selected days were randomly chosen with the proviso that they were business days.

Table 5.4. The data available from 28 smart meters were separated into two subsets. From each of the 15 training meters, nine daily time series were randomly extracted to build up the training dataset. From each of the 13 testing meters, 19 daily time series were selected to make the testing dataset.

		Household	**Industry**	**Services**	**Utilities**	**Transportation**	**Total**
Meters	Train	5	3	4	2	1	15
	Test	4	3	3	2	1	13
Dataset	Train	45	27	36	18	9	135
	Test	37	57	57	38	18	207

The resulting decision tree—Figure 5.6—is fairly simple. Four shapelets are almost enough to classify the training dataset using five labels.

The shapelet I splits the dataset into two groups: one for household and transportation consumption, and the other for services, utilities, and industry. As can be seen in Figure 5.7, this shapelet captures a peak in the evening as a pattern that favors “household” and “transportation” classes. Furthermore, the shapelet II splits these two labels—Figure 5.8a. Since rush hour occurs twice a day, just before and after traditional work hours, it is natural that electric transportation consumes more energy during peak commuting hours.

On the other hand, shapelet III separates “industry” from the “services” demand series. In addition to the rise in the demand of both areas during the morning, and the decrease in the afternoon/evening, there is a break around lunch, which occurs only for industrial demand. Note that the descent phase of the demand for “services” extends into the evening.

Finally, the energy demand by industry and utilities presents different patterns. Utilities demand has a smoother fall in demand during the night compared to industry, and it does not get a break in demand towards lunch.

In conclusion, shapelets were able to identify the main patterns for each type of consumption, and they are quite interpretable from a domain knowledge point of view. However, the decision tree was evaluated against the training dataset and achieves only 60% accuracy. The low number of meters available may have biased the performance of shapelets in this scenario.

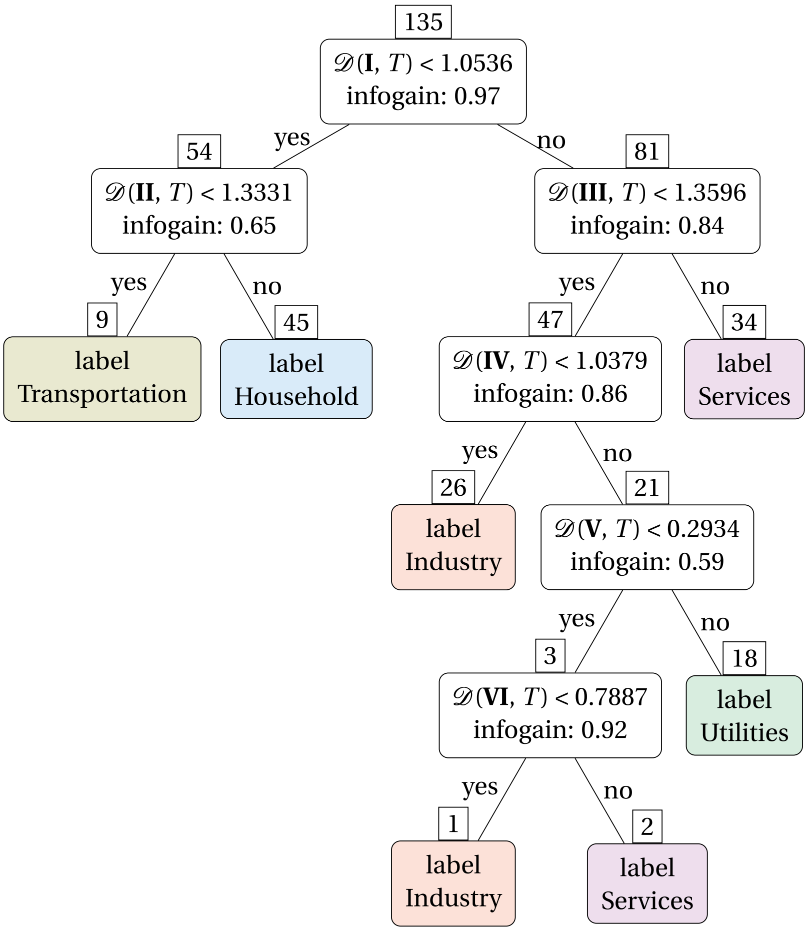


Figure 5.6. The decision tree classifies the type of power consumption.

5.6 Conclusions

Shapelets technique was applied to power load time series collected from three different hierarchical levels of the grid (national, primary power substations, and secondary power substations). Four use cases were defined considering the potential application of shapelets to answer a specific goal. This study is the first to analyze the impact of this technique in more than one use case in the energy domain, following the same methodology and at more than one grid level, while others focus on a single aspect and at a specific grid level. That was made possible by the access to real private data collected in the Portuguese smart grid and the real analytics challenges that were posed, establishing the hypothesis that power load curves alone might have inherent information capable of solving the challenges.

Shapelets have been shown to be useful for the study of pattern extraction and knowledge to understand human activities as energy consumers. This reinforces the importance of data-driven services in the energy sector that can monetize (extract additional value from) data collected within smart grids. The inability to recognize the value of the data in addition to the siloed application in which the data are collected have been one of the barriers identified,

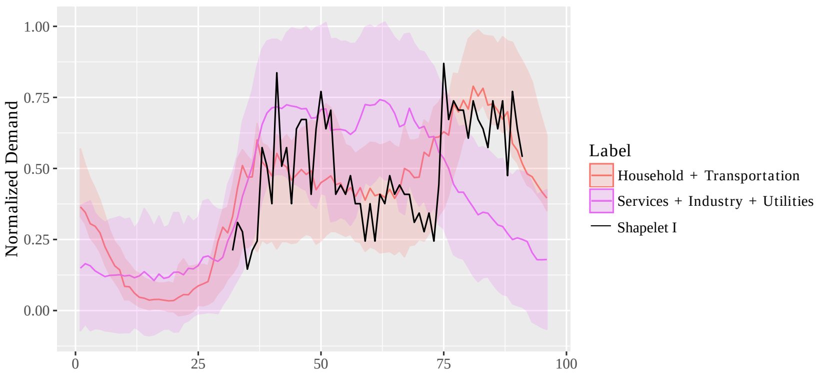
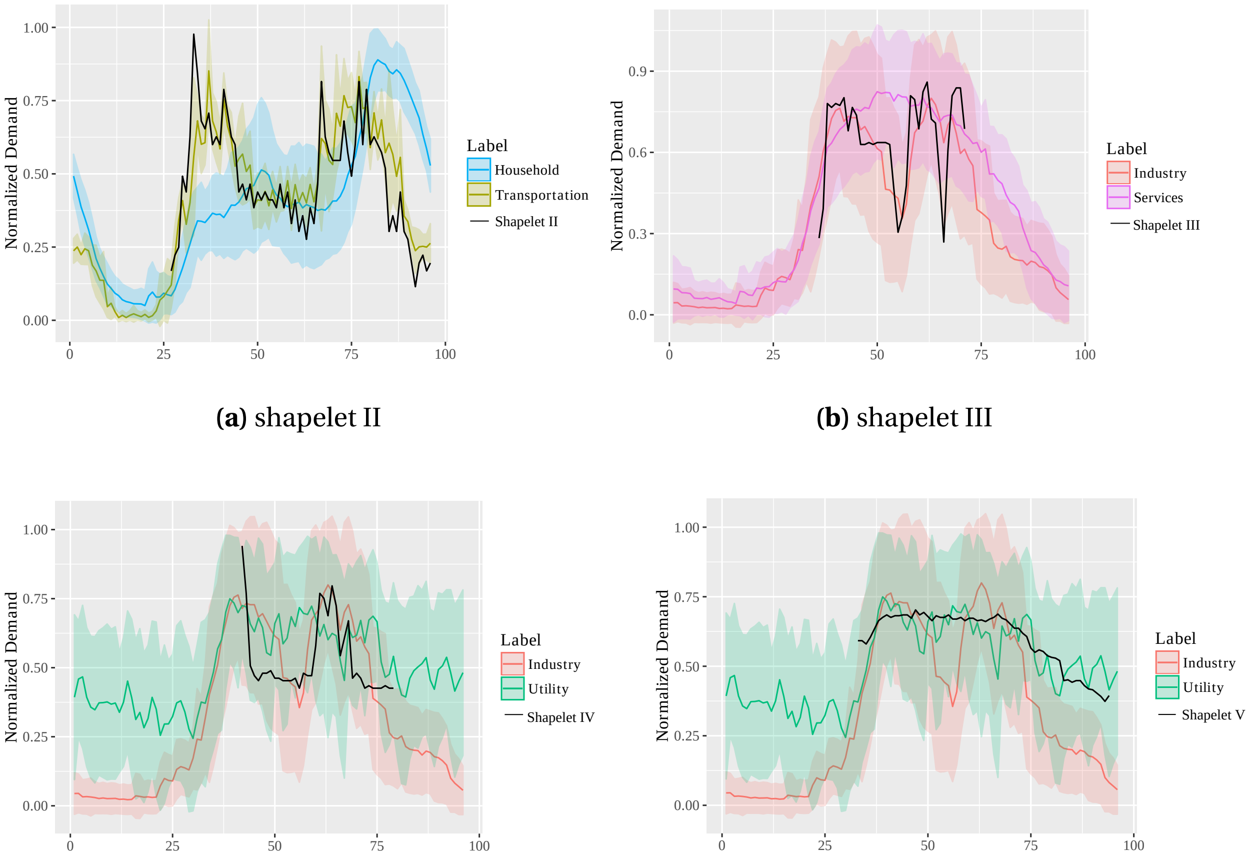


Figure 5.7. Normalized demand of two groups detached by shapelet I.



(a) shapelet II

(b) shapelet III

(c) shapelet IV

(d) shapelet V

Figure 5.8. Normalized demand of classes detached by shapelets. The shapelet II detaches "transportation" from "household" through the demand patterns in the morning and evening. The shapelet III separates "industry" from "services" by a demand break around lunch time. The shapelet IV and V splits the time series into two groups: "industry" and "utility".

as outlined in Section 5.1. Electrification (heating and mobility as examples) and integration of more variable renewable energy (VRE), towards the ambition of 100% renewable electricity, also depend on knowing the purpose of energy consumption and whether there is flexibility on the demand side and capacity of the grid to support those two goals. The fourth use case demonstrates the possibility to classify the purpose of energy consumption and which repeatable patterns the load follows beyond the general patterns highlighted in many studies at the system level. The use case in this chapter is more meticulous in the study of consumption at the primary and secondary substation level.

In particular, shapelets have shown adequate accuracy in all four use cases:

- Weekends—A shapelet was enough to distinguish the daily load time series of weekends from the business days with an accuracy of 97% (Section 5.5.1). The interpretable shapelet is placed in the middle of the day, and grasps the demand break during lunch, a trend that exists on business days but not on weekends.

- Early Monday Morning—The resulting decision tree has a high accuracy of 89% (Section 5.5.2) capable of classifying the daily load time series as weekends, Mondays, and other working days. The two most important shapelets are: (i) a similar shapelet as the previous case with the same goal, the weekend group time series, and (ii) a shapelet that occurs in the early morning on Monday morning, with lower demand—a deeper valley in the curve—when compared with other business day time series. Compared with [104], we used data from the system level (instead of 33 buildings at the low-voltage level) with the same number of quarter-hourly training time series (namely 300 compared to 272) and a higher number of testing time series (namely, 2566 compared to 636). Although they have slightly different goals, both studies reach the same degree of accuracy (89%).

- Load Dynamics in Substations—The resulting decision tree has a high accuracy between 83% and 90% for the two testing datasets (Section 5.5.3), capable of identifying the substation load state, due to scheduled maintenance, failures or maneuvers, even that the load curve presents a high amplitude change by a recent maneuver.

- Type of Power Consumption—The resulting decision tree has an accuracy of 60% (Section 5.5.4), capable of classifying daily consumption according to the type of consumption without any other data beyond the own load curve. The amount of data available for testing may have biased the performance of shapelets in this use case. Once again, the shapelets are interpretable and reinforce the tacit and empirical knowledge practitioners might have about the consumption patterns present in different types of consumption. Compared with [105], we used data from 28 substations (instead of 10), with load curves detailed by 96 datapoints—quarter-hourly (instead of 24 datapoints—hourly) to classify them into one of the five classes (instead of two classes). Our accuracy results reflect the classification ability of the features extracted only by shapelets, while the other study combines the shapelet features with other features to build the classifier, preventing one from pondering shapelets alone).

Thus, this study extends the literature on shapelets applied to the energy domain. At the same time, the study has some limitations, including relying solely on the shapelet technique. However, such a decision was made because of the desire to study the potential of this technique and to reinforce that the power load time series has value beyond the primary functions for which those data are collected. The second limitation is that one relies solely on a database of Portugal power load time series. This is due to the lack of real public data collected within smart grids.

Moreover, the resulting shapelets, and the decision trees that utilize shapelets while node splitting, are interpretable. Interpretability is one of the crucial characteristics in the context of electricity companies in a high-regulated business in Europe, which seeks to regulate AI across several applications, particularly models that act on or support decisions over critical infrastructure. Interpretability eases the application of data-driven services once it must be approved by managers, understandable by grid operators, and defensible before the regulator. Additionally, the interpretability of shapelets increases the knowledge of domain practitioners and, cyclically, the fact that their domain knowledge is "recognized" by the model reinforces its acceptance and adoption. Shapelets fall into the category of techniques that deliver interpretable results.

As final remark to reinforce the imperative to grid operators to boost their progress toward being more data-centric and to provide load curves in higher detail. This promotes research and development on innovative technologies, and accelerates the adoption of new impactful energy models, such as the integration of more VRE, development of generation and demand side flexibility, integration of more decentralized energy resources (DER), electric vehicle charging and other electric vehicle models like vehicle-to-grid (V2G) and vehicle-to-anything (V2X), microgrids, energy communities, and peer-to-peer energy (P2P energy). There are questions as to how these various energy models work together toward the energy transition and how they would be integrated into an existing grid that wants to be resilient and sustainable, both economically and environmentally.

Chapter 6

Cluster-based Load Forecasting

6.1 Introduction

Cluster-based methods are suitable for improving time series forecasting. It is possible to further improve the accuracy of the forecast by training the models with subsets of the time series that behave similarly.

Taking into account the results of Chapter 5 and Section 3.5, load time series can be clustered into groups according to the patterns presented in its own datapoint sequence. Instead of training individual models, one model is trained per cluster with time series from different assets but with a “level of similarity”. This methodology has the following advantages:

- Fewer models to train and store compared to Section 4.1 might be relevant if those predictors must be retrained periodically;
- More data available to train and test within each cluster, and thus higher complex models, meaning more free parameters to fit, might be considered;
- More data means more viability to consider ensemble techniques;
- Cross-fertilization learning between assets is possible, and that might benefit overall load prediction, when compared with Section 4.1 in which models were fitted individually;
- When a new asset is deployed or no existing historical data are available to train individual models, manual or automatic selection of the cluster to which the asset would belong might be an alternative approach instead waiting the availability of data enough to train the individual model.

6.2 Related Work

In fact, time series clustering is a subject that has been applied in forecasting using different approaches [96, 117].

Geva utilizes subsequences clustering which aims to group regimes (stationary subsequences of time series) [118]. The method follows four steps:

1. The time series are rearranged into a set of sequences extracted by a continuously sliding window and optionally applying feature extraction on each window;

2. Hierarchical unsupervised fuzzy clustering is applied on the extracted subsequences set, resulting in temporal patterns grouped together and called regimes;

3. Models are fitted with the subset data of each regime/cluster; and

4. Forecasting is performed by a combination of all model outcomes weighted by the degree of membership of the last temporal subsequence in each of the cluster, meaning how it fuzzily matches with each regime/cluster.

Although the author demonstrated the approach to overcome the general non-stationary nature of time series, it is important to point out that clustering subsequences extracted from sliding windows (streaming subsequences) are considered meaningless in the clustering exercise, as defended by Keogh, Lin, and Truppel [119].

As power load curves show a daily, weekly, and annually seasonality, several authors who applied similar approaches have chosen the 24 hours non-sliding window to split time series into (daily) subsequences with the aim of improving the accuracy of time series forecasting [120–122] or improving hierarchical aggregated forecasting [123–125]), with K-Means [126], hierarchical clustering [120], KNN [121], Fuzzy C-Means (FCM) [125], K-Shapes [123] or Functional High Dimensional Data Clustering (funHDDC) [122]. The raw subsequences are usually normalized and reduced using Min-Max or Max-Abs scalers accompanied by PCA reduction.

Another work by Martínez-Álvarez et al. makes use of clustering as a means of improving time series forecasting [122]. It follows a strategy based on pattern sequence similarity, which was originally developed for discrete time series. The approach assumes that repeating patterns can be discoverable and evaluated in their immediate future. Hence, the strategy applies a discretization of time series using the symbols founded by clustering subsequences. Later, it retrieves the sequence of labels that occur just after the sample is forecasted. This sequence is searched within the historical data, and every time it is found, the sample immediately after is stored. Once the search process is completed, the output is generated by weighting all stored data.

Alternatively, model-based clustering has also been applied as hybrid clustering scheme and pattern recognition [127, 128], subsequence clustering based on Hidden Markov Models [129], and dynamic clustering over time [130].

6.3 Method

Consider the power load as a real-valued discrete time stochastic process

$$Y = \{Y(t) : t \in \mathbb{N}\}. \tag{6.1}$$

One is interested in the evolution of this process in the future. If the process Y was observed over the interval $[1, T]$, one would like to predict the behaviour of Y on the entire interval $[T+1, T+\delta]$, where $\delta > 1$.

Given the existence of n independent stochastic processes, one for each asset in which the power load is measured. Let the matrix $\mathbf{Y}$ represent the measured power load,

$$\mathbf{Y} = \begin{bmatrix} y_{11} & \cdots & y_{1n} & \cdots & y_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{t1} & \cdots & y_{tn} & \cdots & y_{tN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ y_{T1} & \cdots & y_{Tn} & \cdots & y_{TN} \end{bmatrix}. \tag{6.2}$$

Each entry y_{tn} denotes the power load of asset $n \in [1, N]$ at time $t \in [1, T]$ and the vector $\mathbf{y}_n$ all the entries of asset n (the n-th column of matrix $\mathbf{Y}$).

From matrix $\mathbf{Y}$, divide all time series $\mathbf{y}_n$, $\forall_n$ into K clusters, say $\mathscr{C}_1, \mathscr{C}_2, \cdots, \mathscr{C}_K$, containing load curves with similar patterns. The goal is to cluster time series that share temporal patterns. Due a considerable amount (roughly $N \approx 100\,000$) of 6 years long time series (roughly $T \approx 210\,000$ datapoints each), one will consider feature-based approach capable to convert raw time series into a feature vector of lower dimension. Note that the alternatives — model-based approaches — usually have scalability issues [131] or — shaped-based approaches — imply the use of raw time series and distance measures, as DTW, along the whole long time series.

Choosing an appropriate data representation method is considered as the key component which affects the efficiency and accuracy of the solution, as the same time promoting computing performance. Consider the function $\Phi : T \to U$ that extracts features from a time series $\mathbf{y}_n$. A purposed cluster-based discretization function transforms a numerical time series belonging to set T into a sequence of symbols belonging to set U. Not only does this discretize power load time series, but it also reduces data assuming that the features designed to distinguish time series are held. Section 6.4 describes in detail the purposed cluster-based discretization.

With the feature extraction function and the adequate distance function $d : U \times U \to \mathbb{R}$, it is viable to use simple clustering techniques such as partitional or hierarchical clustering to rearrange time series into groups.

Let consider $\mathbf{Y}^{(1)}, \cdots, \mathbf{Y}^{(K)}$ as the resulting matrices of grouping time series into K clusters. Each column $\mathbf{y}_n^{(k)}$, which belongs to cluster $\mathscr{C}_k$, denotes the dependent variable and has associated explainable variables $\mathbf{X}_n$ (see Equation 4.2). This dataset is used to fit non-individual regression models $f^{(1)}, \cdots, f^{(K)}$, one per cluster k instead of one per asset n, as done in Chapter 4. Generically, the individual disaggregated load model is the particular case with $K = N$, the number of assets, and the one-size-fits-all disaggregated load model, $f^{(G)}$, the particular case with $K = 1$. Figure 6.1 outlines the method. Section 6.5 covers the decisions and results of the clustering process, whereas Section 6.6 the forecasting results of cluster-based models.

In the end, all models are evaluated considering the same error metrics. Using the same regression structure, it is empirically expected that the error of individual models is better than cluster-based one or the one-size-fits-all model in the vast majority of cases. However, maintaining a few K regression models is easier than maintaining thousands of models, as other advantages mentioned in the previous section. If the error metrics are satisfactory for cluster-based models, it is reasonable to consider improving with further complex regressors with augmented forecasting effectiveness, something viable due to the existence of more datapoints to train with and fewer models to compute and maintain.

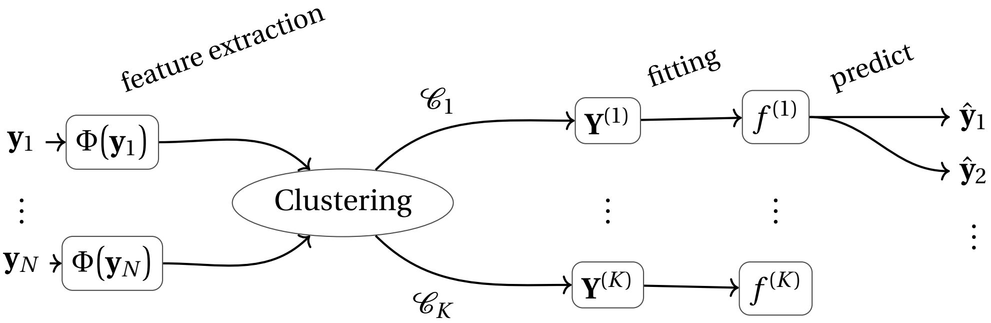


Figure 6.1. Features are extracted from load time series before applying clustering. The datapoints of each cluster are together used to fit a regressor which, in turn, is used to forecast cluster's assets consumption.

6.4 Cluster-based Discretization

In this case, the discretization function aims to keep and emphasize the characteristics that the clustering process would use to distinguish and group time series, and also to remove unnecessary information, as the function is also a data reduction process.

To design the function, it is important to highlight the characteristics discussed in previous sections which set the reasons of defining a new feature extraction:

- The power load time series features daily, weekly, and yearly seasonalities, as exhibited in Figure 3.1, and patterns are held by these seasonalities;
- The GAM models, as defined by Equation 4.4, exploit the lagged load of the last day or week as explanatory variables, part of autoregression. This is regulated by the existence of a high degree of stationarity in the short term or within the called regimes;
- Notably, stationary regimes change throughout the year cycle. For example, summer holidays in August may exhibit a regime that differs from other periods, as well as specific public holidays and season changes, as shown inn Figure 3.1;
- The same regimes and patterns do not feature in all time series. For example, daily patterns differ in different types of power demand, as shown in Figures 5.7 and 5.8.

The purposed cluster-based discretization function $\Phi : \mathbb{R}^{\alpha} \to S$ is designed to transform a real-valued time series into a symbol sequence, extracting the daily pattern in the time domain. Note that in contrast, the well-known symbolic aggregate approximation (SAX) technique is performed only on the basis of mean value in time domain. To transform a long time series, the function Φ is applied through a fixed tumbling window aligned with the day. In practice, a sequence of 48 half-hours ($\alpha = 48$) is discretized into a symbol $s \in S$. The dictionary of symbols, S, also known as the set of standardized daily patterns, is achieved by a previous step, which involves extracting snippets.

6.4.1 Snippets

In the context of the time series summarization problem, time series snippets is a technique to extract the "representative" subsequences of a long time series [101]. It is a better technique than the other obvious definitions: motifs, shapelets, cluster centers, or random samples. While motifs reward the fidelity of conservation, snippets also reward coverage. Informally, coverage is some measure of how much of the data is explained or represented by a given snippet. Shapelets are defined as subsequences that are maximally representative of a class, as used in the classification problem in Chapter 5. Shapelets are supervised, snippets are unsupervised. Shapelets are generally biased to be as short as possible. In contrast, one wants the snippets to be longer, to intuitively capture the "flavor" of the time series.

The snippets technique is applied for each time series $\mathbf{y}_n$ in parallel with a fixed length and a constant number of snippets to extract. In order to obtain daily patterns, a length of 48 datapoints is fixed, and the time series was wrangled to start and end exactly at midnight, and the exceptional days, in which daylight saving time events occur, were removed. After testing the technique with different values for the number of snippets parameter, four snippets would be enough for the majority of cases. Figure 6.2 shows the four snippets resulting from a power load time series example.

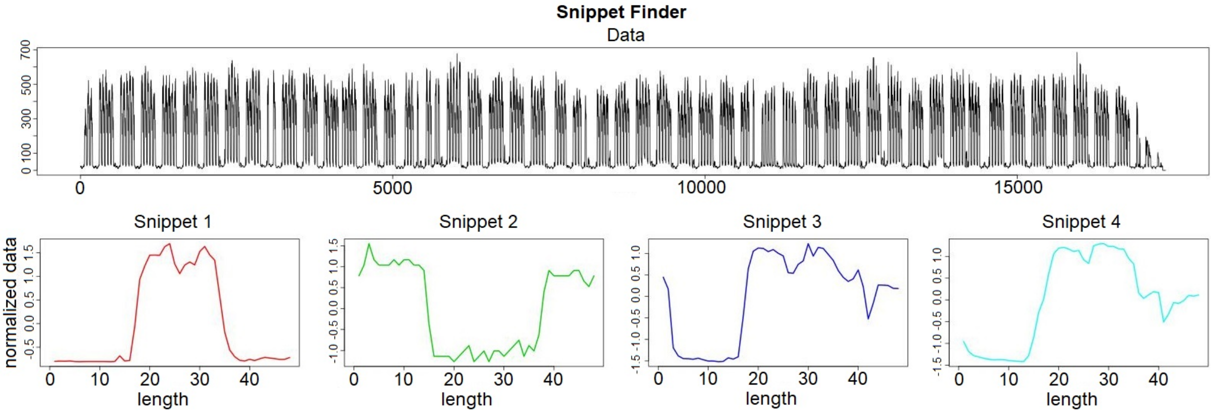


Figure 6.2. The resulting snippets for a power load time series (from the PTC set). The plot above shows the original power load. The four plots below are the snippets, z-score normalized. Snippets 3 and 4 are mostly for working days. Snippets 1 and 2 are mostly for weekends and August, though very similar. Note that the amplitude is not considered to match the snippets with the daily power load time series. Although all weekends and August working days match with snippet 1 from a shape perspective, they have very different amplitudes.

6.4.2 Dictionary of Symbols

The composition of the dictionary of symbols, that is, the codomain S of function Φ, aims to collect all the daily shapes that stand out from the time series of power loads. The snippets, as the "representative" shapes of each time series, are just a first step that led to a set of $4N \approx 400\,000$ shapes. To obtain the final set of symbols, a partitional clustering algorithm is applied. As the snippets are already scaled and the time domain is aligned among the snippets, meaning the i-th element of all snippets refers to the very same half-hour of the day, the choice simply falls on k-means with the Euclidean distance. Two assessments contribute to determining the optimal value of k shapes: for each k, (i) the evaluation of silhouette score [132] and (ii) the domain interpretation of resulting shapes.

The resulting dictionary of symbols $S = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q\}$ is shown in Figure 6.4 in which examples of snippets are grouped in the respective symbol cluster, and the average shape $f(s)$ is colored. The silhouette score is calculated for each k and the higher the value, the better the consistency of the cluster. The k with the best silhouette score was not chosen because few symbols would be defined, unsatisfactory to discretize time series. It was decided to have more symbols that balance between the silhouette score and the domain interpretation. A $k = 17$ was decided that results in $|S| = 17$ symbols and the silhouettes of the resulting cluster are shown in Figure 6.3. To exemplify the domain interpretation of resulting symbols to access the best k, consider the following statements:

- The symbol d has a peak in the evening, as the pattern of household class considered in Figure 5.8a.
- The symbol e follows the same pattern as the transport class considered in Figure 5.8a.
- The symbols $\{f, g, h\}$ have a consistent demand throughout daylight, though with small differences in the location of the peaks. They follow the same patterns as the services class considered in Figure 5.8b.
- The symbol h follows the same pattern as the industry class considered in Figure 5.8c.
- The symbol f follows the same pattern as the utility class considered in Figure 5.8c.
- The symbols $\{o, p\}$ follow the duck curve due to local photovoltaic generation [133] or due to use mainly for nocturnal illumination depending on the actual amplitude. $\{l, m, n\}$ follows similar pattern with more prominent peaks.

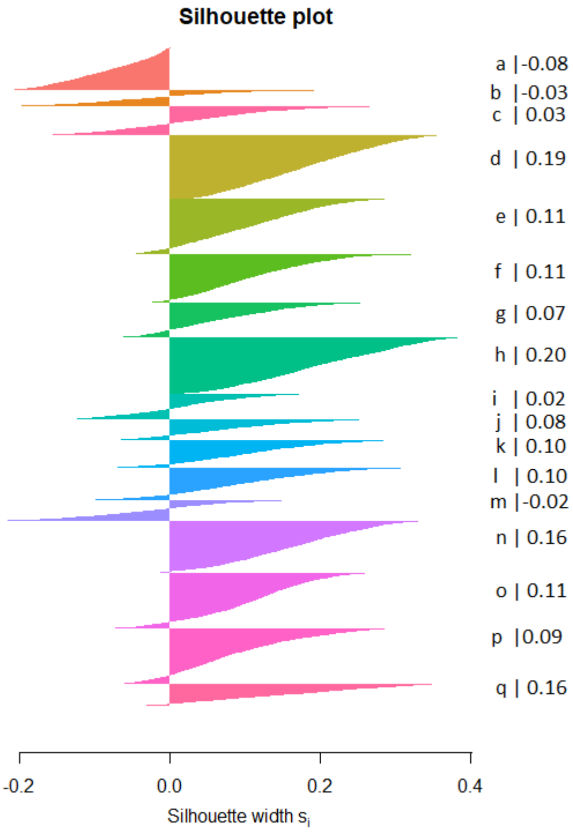


Figure 6.3. The silhouette widths for each cluster. Note that the silhouette of symbol $a \in S$ was expected to have a low silhouette because it also assimilates the noise subsequences that become scattered within the cluster frontiers. The higher the silhouette value, the better the consistency of the cluster, and the lower the suspicion of mismatching the point between neighbor clusters in the Euclidean space.

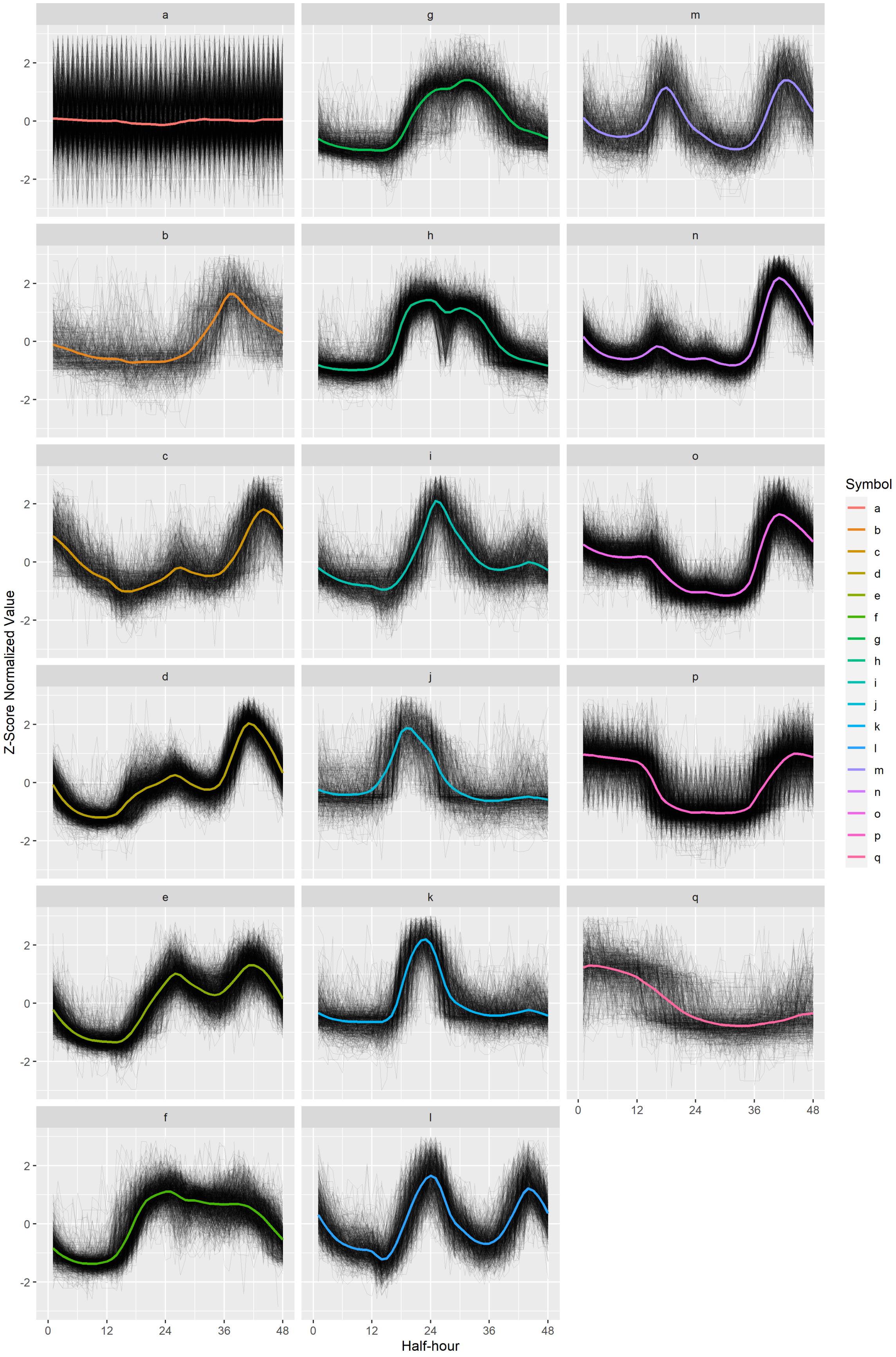


Figure 6.4. The dictionary of symbols S that stand out from the time series of power load, after clustering snippets.

6.4.3 Discretization

Given the dictionary of symbols S, the cluster-based discretization function Φ is finally defined, explaining how the match is performed between a daily sequence of real values and the respective symbol s. The function looks for the minimization of the Euclidean distance between the z-score-normalized input and each possible symbol,

$$\Phi(\mathbf{y}) = \arg\min_{s \in S} d_{\text{EUCLIDEAN}}\big(z(\mathbf{y}), f(s)\big) \tag{6.3}$$

where d and z are, respectively, the Euclidean distance and z-score scalar functions, and $f(s)$ the numerical average shape of the respective symbol.

As an example, the same time series used for Figure 6.2 is discretized into symbol sequence and plotted as a calendar in Figure 6.4. The time series obeys a specific pattern during the working days, except on weekends, the last two weeks of August, public holidays, and the days around them. In this scenario, the long weekend, meaning the days off occurring between the public holiday and the weekend, affects the shape of power load as on the Monday and Tuesday Carnival week or the Thursday's Corpus Christi holiday and the respective Friday off. Furthermore, the effect of the holiday electricity demand on Thursday extends to Saturday, as can be seen in the plot. Christmas and the summer holiday season are other special periods that impact the demand curve.

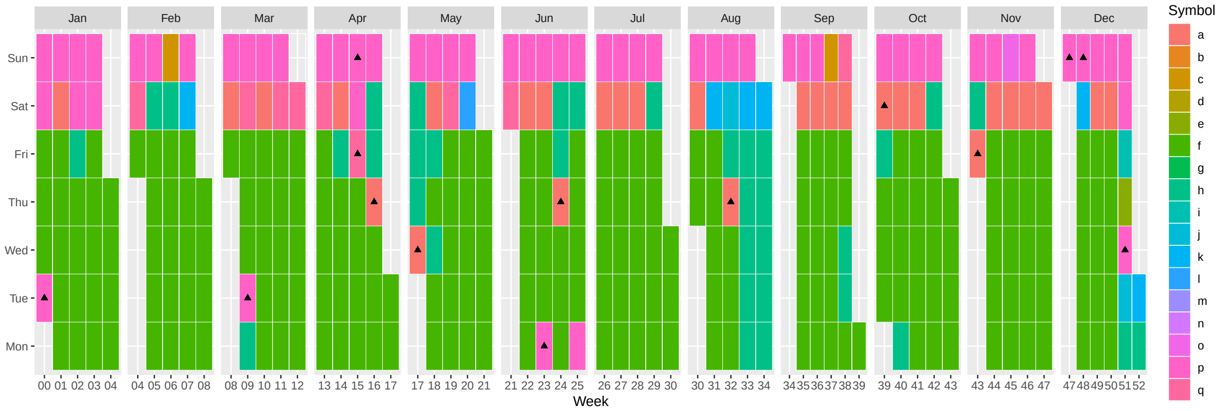


Figure 6.5. The resulting cluster-based discretization of the same time series used for Figure 6.2 plotted as a calendar, temporal portion of 2019. The time series obeys a specific pattern during working days, except on weekends, the last two weeks of August, public holidays, and the days around them.

6.5 Symbol Sequences Clustering

The clustering divides all time series into groups that contain load curves with similar daily patterns and similar regimes throughout the year. The method exploits the representation of the power load in discrete series to later distinguish and group them. That representation computation is parallelizable and results in the matrix

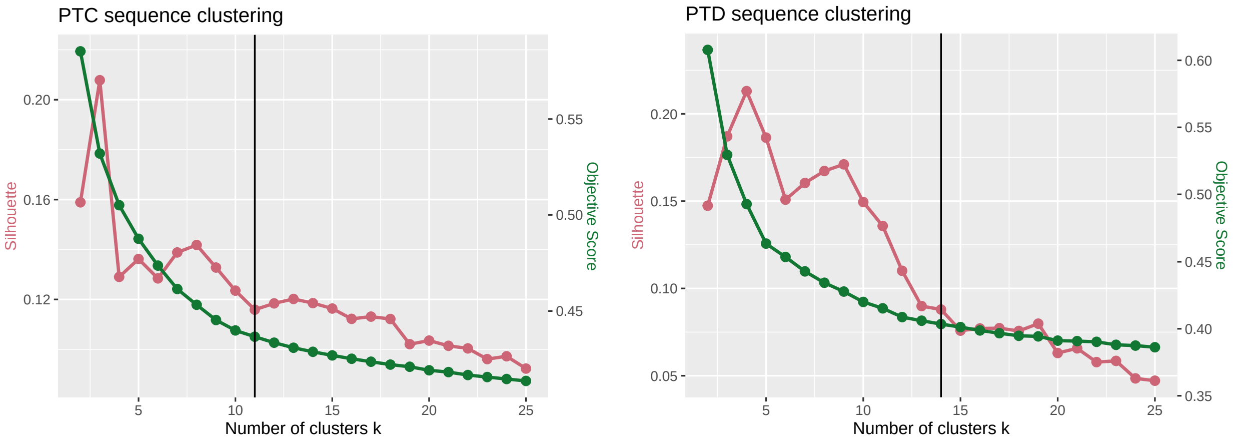
$$\Phi(\mathbf{Y}) = \begin{bmatrix} \Phi(\mathbf{y_1}) & \cdots & \Phi(\mathbf{y_n}) & \cdots & \Phi(\mathbf{y_N}) \end{bmatrix}. \tag{6.4}$$

Indeed, the method proceeds with a manual separation between PTD and PTC time series, naturally having different behaviors considering the role each has in the power grid.

6.5.1 k-Medoids

The k-medoids clustering is selected after some testing with other alternatives, such as hierarchical clustering, k-means, and alternative distance functions. The results were as good as or worse than the final choice for k-medoids and Gower's distance. The sequence is considered in nominal scale and all elements equally weight to Gower's distance[1].

The Partioning Around Medoids (PAM) algorithm carry the greedy search for medoids and three assessments contribute to determining the optimal value of k clusters: for each k, (i) the evaluation of silhouette score, (ii) the evaluation of the objective score[2] of k-medoids using the elbow method, as shown in Figure 6.6, and (iii) the domain interpretation of time series distributed across clusters. 11 PTC clusters and 14 PTD clusters were decided whose silhouette widths per cluster are shown in Figure 6.7.



(a) PTC

(b) PTD

Figure 6.6. The silhouette and objective scores for each k contributes to determining the optimal value of k. As heuristics, the higher the silhouette value, the better the consistency of the cluster. The lower the objective score, the higher the similarity of the observations to their closest medoid.

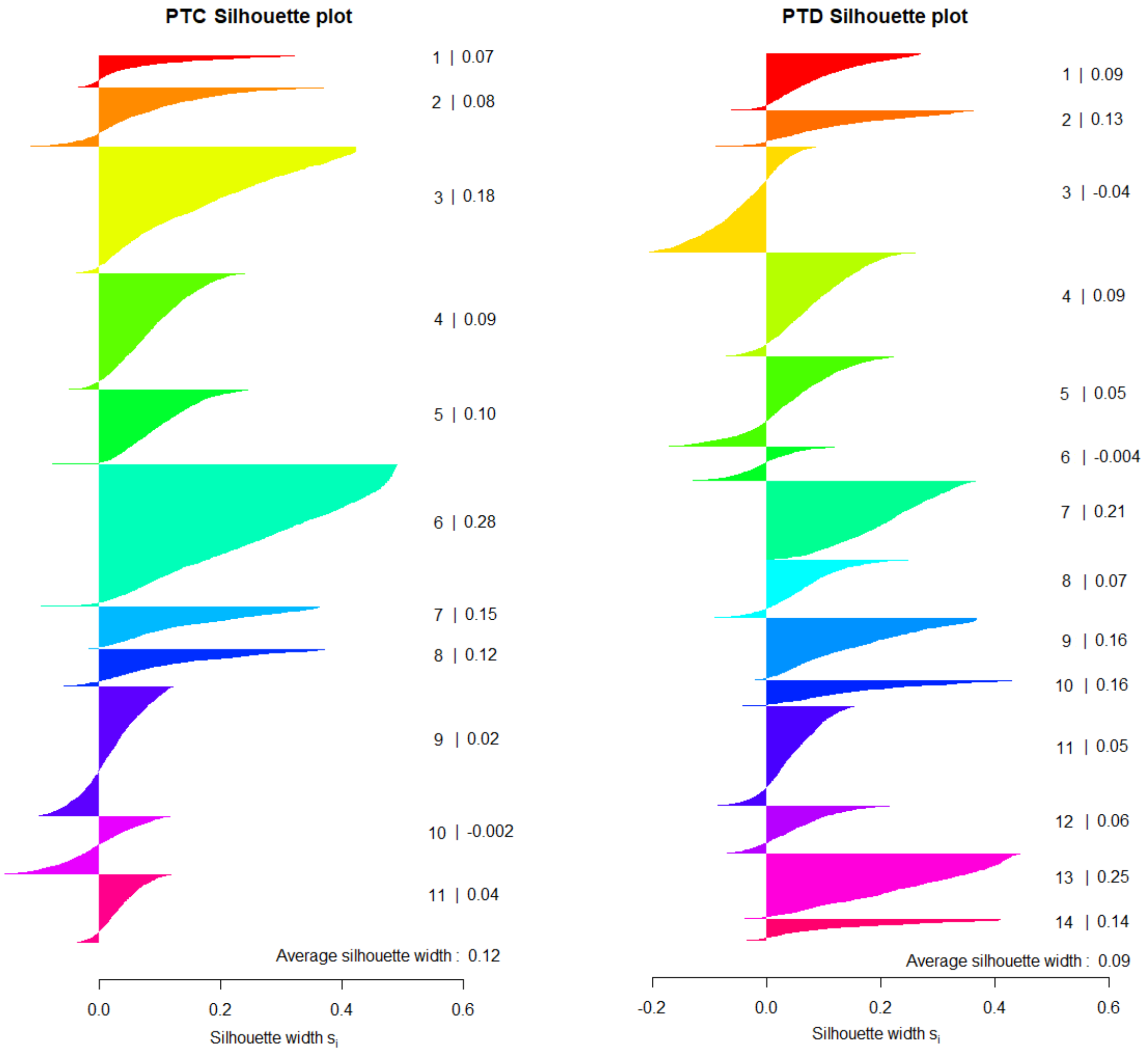
Two additional notes: Due to the high number of sequences, in the case of PTD time series, a 30% random sample was used to find the k-medoids and later the remaining points were grouped by finding the closest representative medoid. Further tests with other random samples of PTD time series have shown the stability of the results regardless of the sample. Another note is about finding the closest representative medoid: when comparing sequences, it is highly relevant that they are aligned to the same year because daily patterns and local regimes throughout the year are partially consequence of holidays and the day of the week on which they fall, which move from year to year. Nevertheless, alternative distance functions could ensure the natural movement of holidays from year to year.

6.5.2 Results

The power load time series are grouped into 25 clusters, $\mathscr{C}_{\text{PTC}\,1}, \cdots, \mathscr{C}_{\text{PTC}\,11}, \mathscr{C}_{\text{PTD}\,1}, \cdots, \mathscr{C}_{\text{PTD}\,14}$, 11 for PTC and 14 for PTD time series. Figures 6.8 and 6.10 introduce the medoids of each

[1] For the particular characteristics of the elements and the Gower's parameters, the distance is liken to Hamming distance after pre-processed inputs with one-hot encoding and before scaling output to the range $[0, 1]$.

[2] Minimize the sum of the dissimilarities of the observations to their closest representative object.



(a) PTC, $k = 11$

(b) PTD, $k = 14$

Figure 6.7. The silhouette widths for each cluster. While the height indicates the size of each cluster, the width shows how each observation contributes to silhouette of its cluster.

cluster. They are the representative *center* of the cluster and are used to determine which sequences belong to which cluster by finding the closest medoid. Naturally, they are interpretable through the figures plotted on a calendar considering months, days of the week, and public holidays and using the same color scheme as in Figure 6.4. Since the time series grouped into the same cluster are not entirely the same, it is important to understand the symbol distribution considering the calendar features. Figures 6.9 and 6.11 exhibit symbol distribution percentage (vertical axis) on each day throughout the year (horizontal axis) by cluster (row) and day type (column). To exemplify the domain interpretation of the resulting PTC medoids, consider the following assertions:

- Clusters $\mathscr{C}_{\text{PTC}1}$ and $\mathscr{C}_{\text{PTC}3}$ do not show an interesting daily pattern, as they follow, respectively, the symbols q and a throughout the year. However, looking at the symbol distribution of $\mathscr{C}_{\text{PTC}1}$ (first row of Figure 6.9), the pattern changes on Sunday for part of the time series belonging to that cluster. This is observable by the symbol distribution that is distinct on that specific day compared to the others.

- All other PTC clusters follow more interesting daily patterns. There are three main dynamics to remark: (i) some clusters reveal stability in the distribution of their symbols throughout the whole year, while others change their daily symbol through a yearly seasonality, (ii) some clusters reveal differences between the daily patterns of working days and weekends, others follow the same daily pattern regardless of the day of the week, (iii) some clusters uncover changes in symbol distribution during August.

- Clusters $\mathscr{C}_{\text{PTC}4}$, $\mathscr{C}_{\text{PTC}5}$, $\mathscr{C}_{\text{PTC}9}$, and $\mathscr{C}_{\text{PTC}11}$ follow the typical curve represented by the symbol h with changes in August for a large part of the time series belonging to these clusters. The difference between them is what happens on weekends and public holidays. The regime of cluster $\mathscr{C}_{\text{PTC}5}$ is stable regardless of weekends and public holidays; $\mathscr{C}_{\text{PTC}11}$ changes its daily pattern for a morning peak curve (symbol j) on Saturdays and public holidays, and during Sundays it goes to a noisy or constant signal (symbol a); the same symbol a occurs on weekends and public holidays for $\mathscr{C}_{\text{PTC}9}$; $\mathscr{C}_{\text{PTC}4}$ shows a load curve mainly for nocturnal illumination (symbol p) during weekends and public holidays.

- Clusters $\mathscr{C}_{\text{PTC}6}$ and $\mathscr{C}_{\text{PTC}10}$ follow a load curve similar to the latter, but the high load extends into the evening (symbol f). $\mathscr{C}_{\text{PTC}10}$ changes their daily patterns during weekends and public holidays to symbols p or q, while $\mathscr{C}_{\text{PTC}6}$ do not.

- Cluster $\mathscr{C}_{\text{PTC}2}$ follows the daily symbol g with yearly seasonal changes in the symbol distribution.

- Cluster $\mathscr{C}_{\text{PTC}8}$ follows the symbol e with two maximums during midday and evening. However, a large part of the time series related to this cluster changes its daily pattern during the warm season to the symbol d, which gives a sharper peak in the evening compared to midday.

Moreover, the resulting PTD medoids are also elucidated from the domain perspective:

- Cluster $\mathscr{C}_{\text{PTD}14}$ follows the symbol a throughout the year.

- Clusters $\mathscr{C}_{\text{PTD}1}$, $\mathscr{C}_{\text{PTD}9}$, and $\mathscr{C}_{\text{PTD}10}$ follow, respectively, the symbols f, e, and g throughout the year, as the previously described clusters $\mathscr{C}_{\text{PTC}6}$, $\mathscr{C}_{\text{PTC}8}$, and $\mathscr{C}_{\text{PTC}2}$. However, looking at the symbol distribution of cluster $\mathscr{C}_{\text{PTD}10}$, it is perceptible that a significant part of time series of that cluster has an annual seasonality.

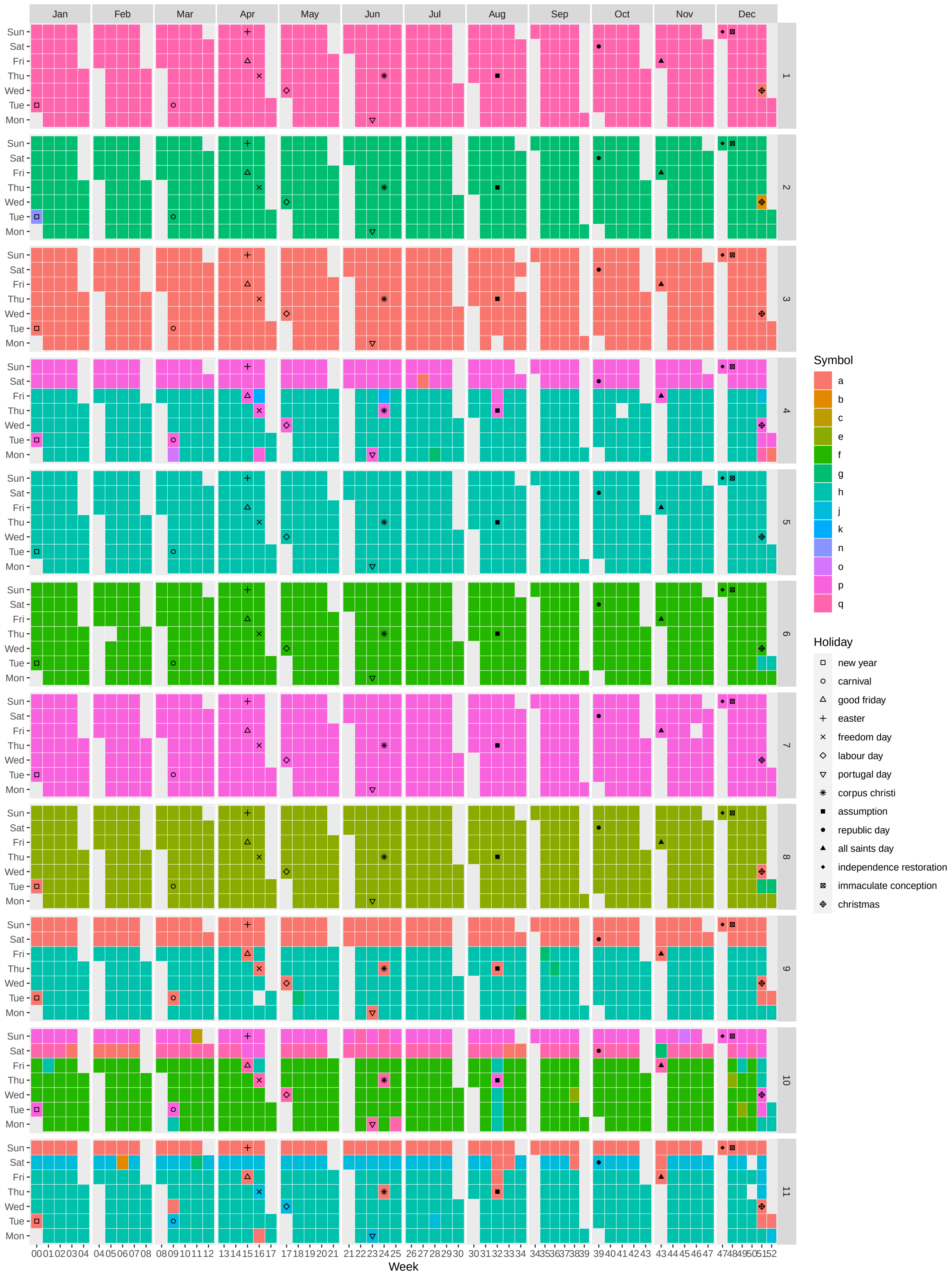


Figure 6.8. The 11 PTC medoids plotted on calendar.

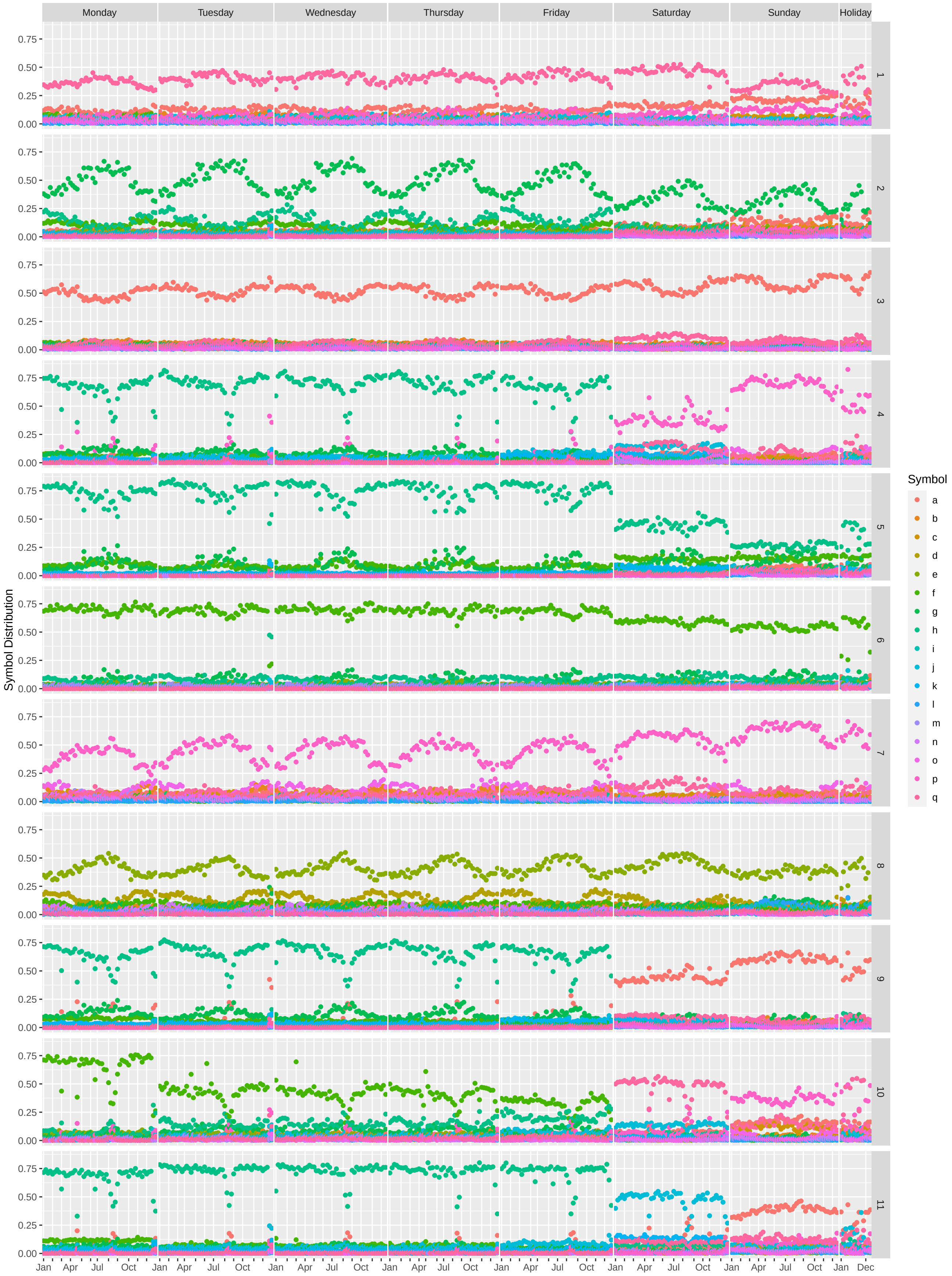


Figure 6.9. Distribution of symbols (vertical axis) on each day throughout the year (horizontal axis) by PTC cluster (rows) and day type (column).

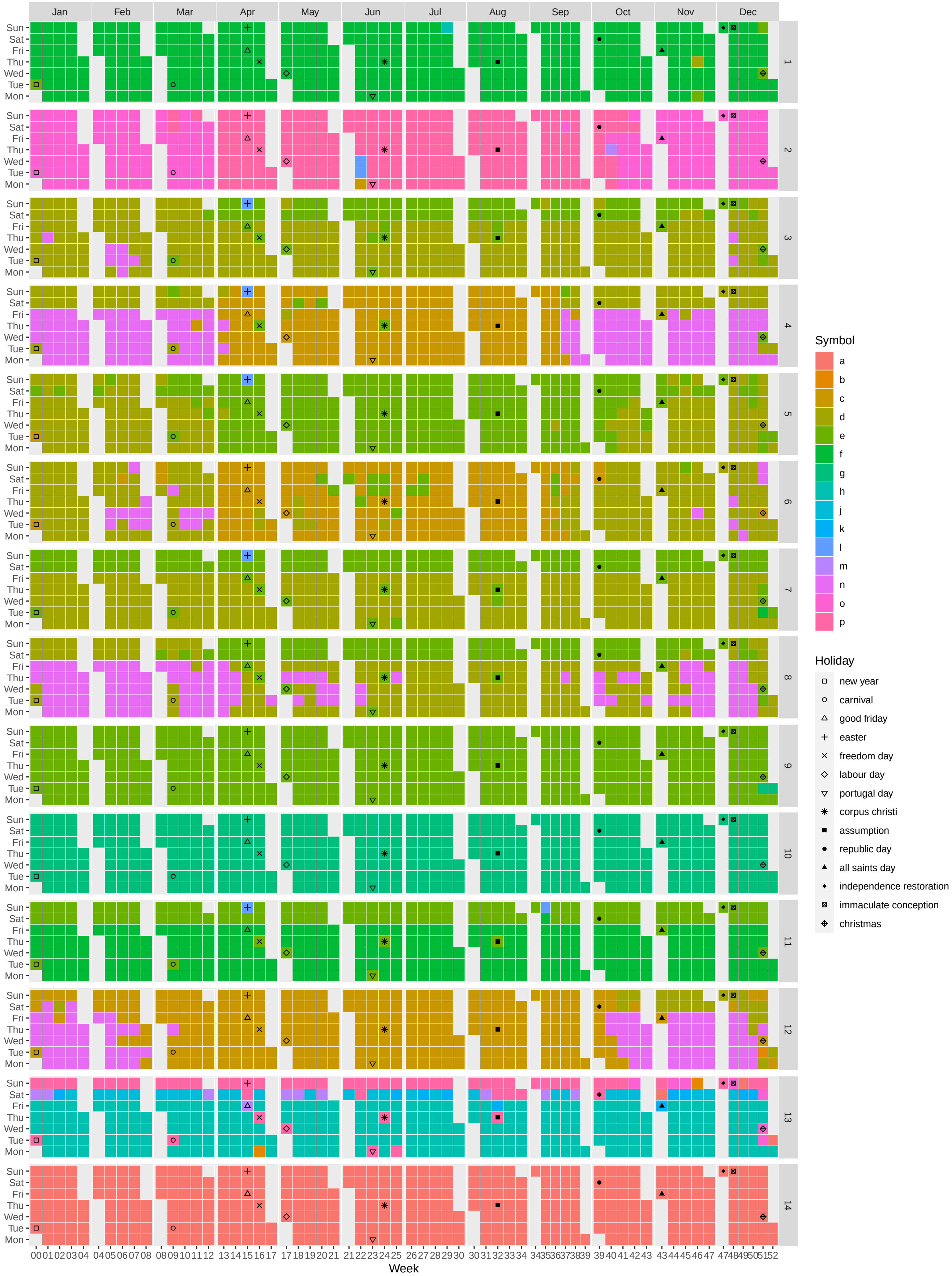


Figure 6.10. The 14 PTD medoids plotted on calendar.

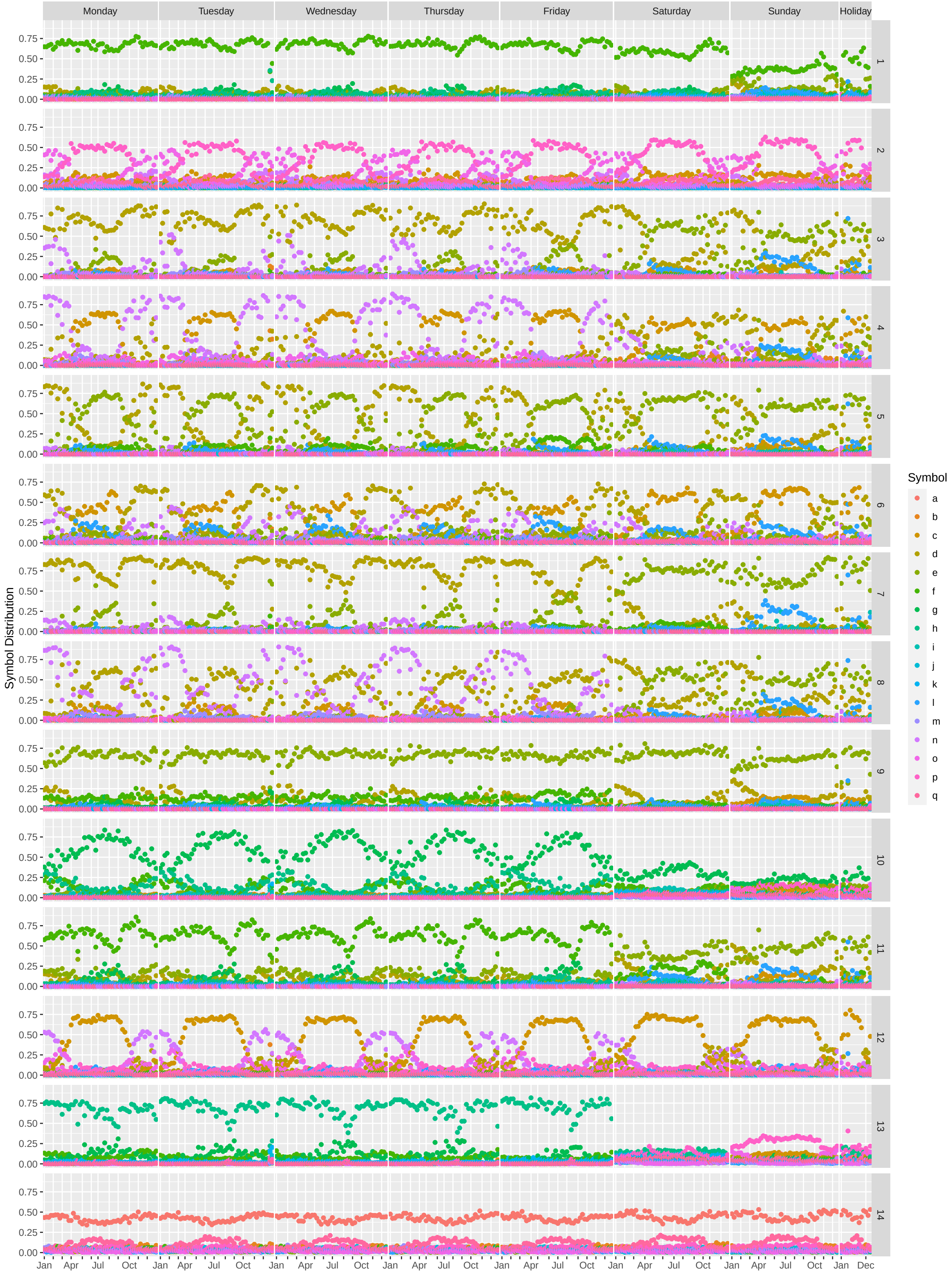


Figure 6.11. Distribution of symbols (vertical axis) on each day throughout the year (horizontal axis) by PTD cluster (rows) and day type (column).

- Cluster $\mathscr{C}_{\mathrm{PTD}\,11}$ follows the same symbol f as $\mathscr{C}_{\mathrm{PTD}\,1}$, but unlike this, the predominant symbol in the cluster $\mathscr{C}_{\mathrm{PTD}\,11}$ on weekends is e. As the reader might verify, these two symbols are similar except for a slightly general decrease in power load in the afternoon. Thus, the two symbols are significantly different because they get the differences between working days and weekends for a significant number of time series.

- Cluster $\mathscr{C}_{\mathrm{PTD}\,13}$ follows the symbol h and changes its daily pattern for a morning peak curve (symbol j) on Saturdays approximating to $\mathscr{C}_{\mathrm{PTC}\,11}$ behavior.

- Cluster $\mathscr{C}_{\mathrm{PTD}\,2}$ follows the symbols o and p.

- The remaining clusters are interpreted with the support of Figure 6.12 in which the four main symbols are ordered in a way that visually unfolds the extents of the regions of cold and warm seasons of each cluster. Despite the evident differences among the clusters, the boundaries between them can start to be considered fuzzy from this k on. In fact, the lower silhouettes shown in Figure 6.7b are from these clusters. Nevertheless, the individual time series are much more complex, and for the sake of simplicity, this is just one aspect regarding the cold and wind seasons.

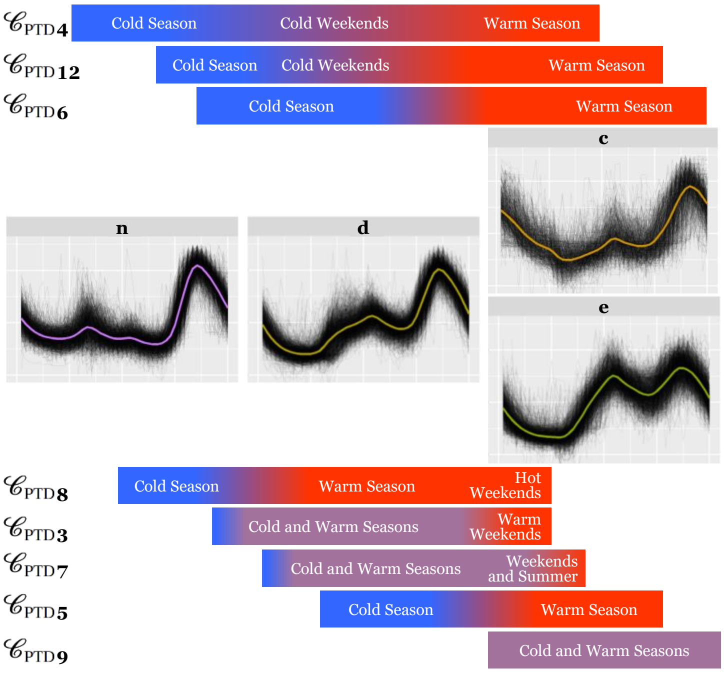


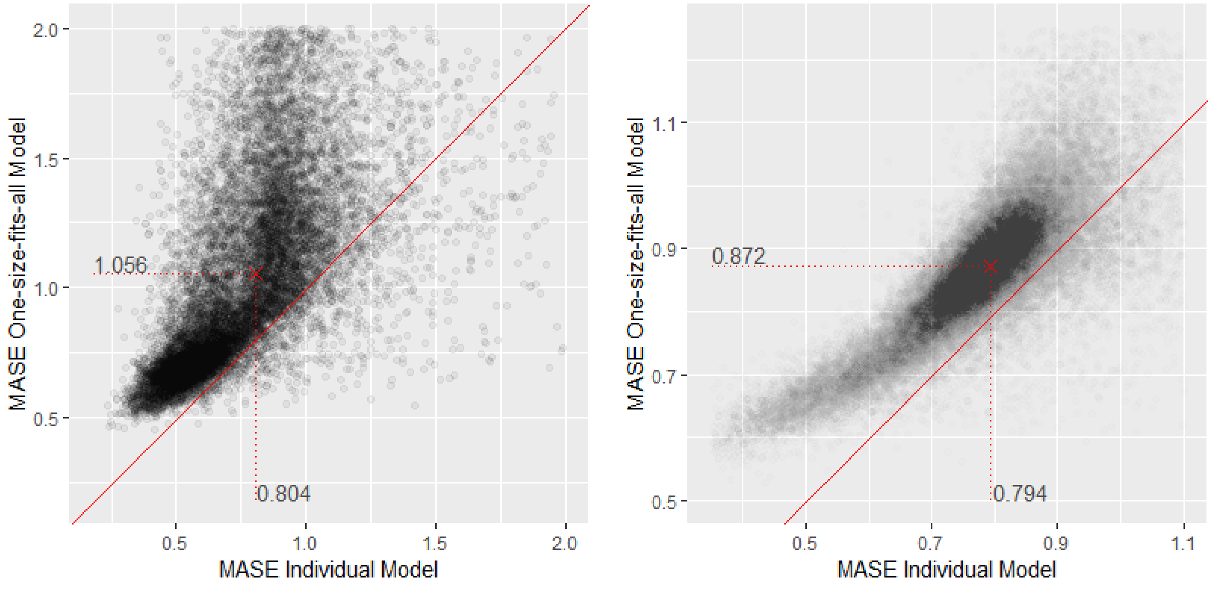
Figure 6.12. The symbols n, d, c, and e are ordered in a way that visually unfolds the regions of cold and warm seasons. The symbol c, which has a smoother upward slope in the morning power demand, unfolds the warm region of the three clusters above, and the symbol e the following.

6.6 Cluster-based Load Forecasting

The cluster-based regression models, $f^{(\mathrm{PTC}\,1)}, \cdots, f^{(\mathrm{PTC}\,11)}, f^{(\mathrm{PTD}\,1)}, \cdots, f^{(\mathrm{PTC}\,14)}$, are fitted with a stratified sample of each cluster dataset. The structure of the model is the same as used in Chapter 4 (Equation 4.3). In addition, disaggregated load forecasting models of one-size-fits-all, $f^{(\mathrm{PTC})}$ and $f^{(\mathrm{PTD})}$, are also fitted using a stratified sample of all PTC and PTD data regardless of their cluster. These two models are the baseline for comparison with the cluster-based and individual models.

The respective fitted model is then used to individually predict the power load of each asset, involving the same dataset, methodology, and error metrics to get comparable results.

The results reported in Figure 6.13 provide further evidence that, for most assets, the individual model is more accurate than the one-size-fits-all alternative. The cloud of points represents the MASE error of the two models for each asset. All the points above the red line mean that the individual model is better qualified and, therefore, justifies the computational effort to individually train each GAM model. Nevertheless, the one-size-fits-all model is better for a small set of assets.



(a) PTC

(b) PTD

Figure 6.13. Scatter plot of MASE error for each individual model and the one-size-fits-all model. Each point is the forecasting error of a specific time series. The points above the red line means that the individual model is better than the latter. The cross mark is the median center of the cloud of points that might be cut.

To pursue the goal of reducing the number of models to train and maintain, or when a new asset is deployed or no existing historical data are available to train the individual model but enough knowledge of what is or would be the general pattern, cluster-based models are a better solution than the one-size-fits-all GAM model.

In general, most cluster-based models achieve better accuracy than the one-size-fits-all model. In Figure 6.14, the cluster-based models $f^{(\mathrm{PTC}2)}$, $f^{(\mathrm{PTC}3)}$, $f^{(\mathrm{PTC}5)}$, $f^{(\mathrm{PTC}6)}$, $f^{(\mathrm{PTC}8)}$,$f^{(\mathrm{PTC}9)}$, and $f^{(\mathrm{PTC}11)}$ are better in terms of the distribution median of MAPE, NRMSE, and MASE when analyzing the pair error metrics of both models.

However, there are exceptions in which the cluster-based model is not better, and in that case, the option is to keep the one-size-fits-all model for the load forecasting of that cluster's assets or apply the same clustering technique to further split the cluster's assets to a new set of groups. Cluster-based models $f^{(\mathrm{PTC}1)}$,$f^{(\mathrm{PTC}4)}$, and $f^{(\mathrm{PTC}10)}$ do not follow that improvement as the other models. Nevertheless, $\mathscr{C}_{\mathrm{PTC}10}$ already presented a worse silhouette width compared to others (Figure 6.7a), and $\mathscr{C}_{\mathrm{PTC}1}$ presented a distribution of symbols in which the symbol q stands out, but is not as dominant as other symbols in the respective clusters (Figure 6.9).

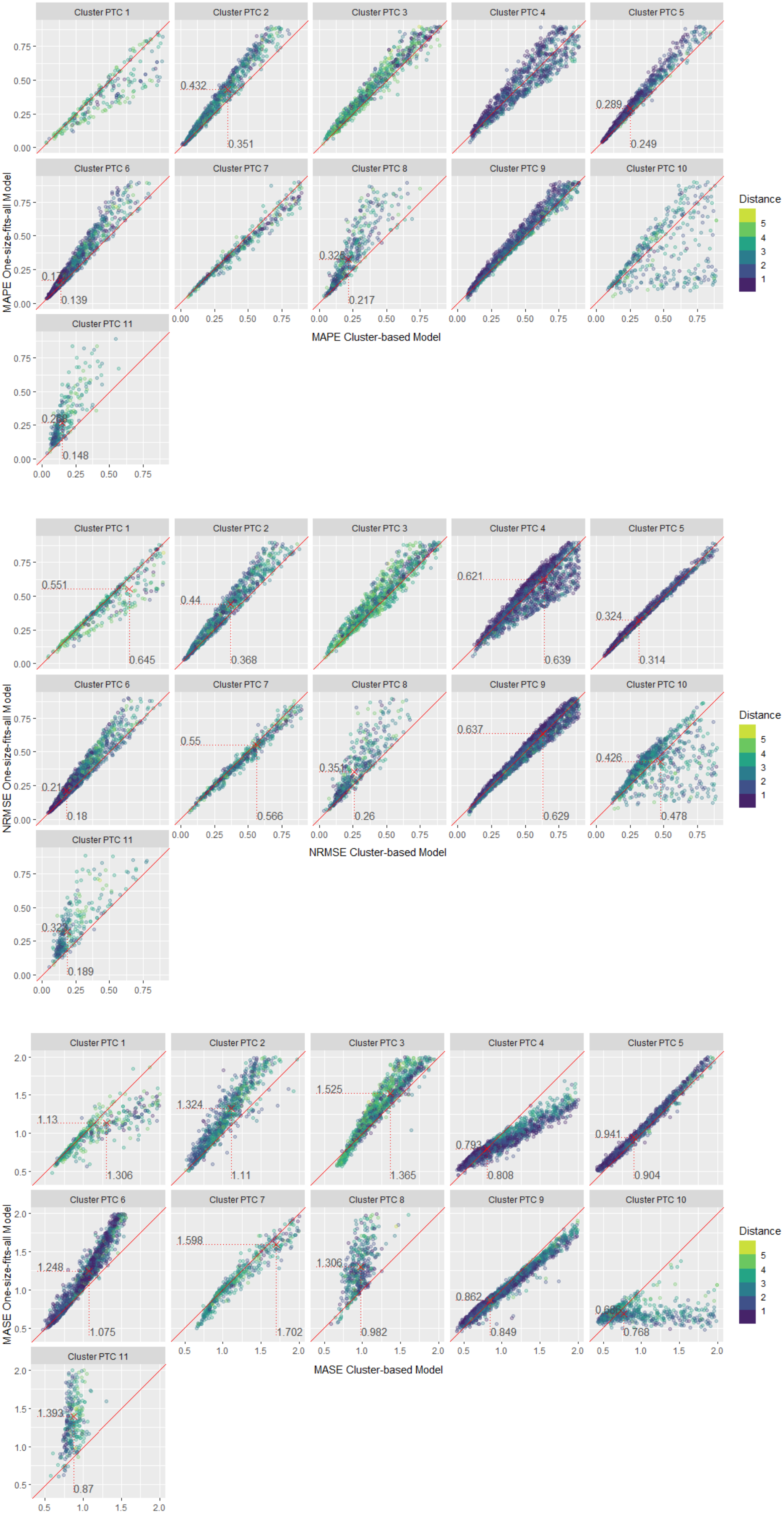


Figure 6.14. PTC scatter plot of MAPE, NRMSE, and MASE errors for each cluster-based model and the one-size-fits-all model. Each point is the forecasting error of a specific PTC time series colored by distance to the respective medoid. The points above the red line means that the cluster-based model is better than the other latter. The cross mark is the median center of cloud of points.

Regarding PTD models, Figure 6.15 shows that cluster-based models $f^{(\mathrm{PTD}\,2)}$, $f^{(\mathrm{PTD}\,3)}$, $f^{(\mathrm{PTD}\,4)}$, $f^{(\mathrm{PTD}\,6)}$, $f^{(\mathrm{PTD}\,8)}$,$f^{(\mathrm{PTD}\,10)}$,$f^{(\mathrm{PTD}\,12)}$,$f^{(\mathrm{PTD}\,13)}$, and $f^{(\mathrm{PTC}\,14)}$ are better in terms of the distribution median of MAPE, NRMSE, and MASE when analyzing the pair error metrics of both models.

The other cluster-based models, $f^{(\mathrm{PTD}\,1)}$, $f^{(\mathrm{PTD}\,5)}$, $f^{(\mathrm{PTD}\,7)}$, $f^{(\mathrm{PTD}\,9)}$, and $f^{(\mathrm{PTD}\,11)}$ have the same median accuracy compared to one-size-fits-all models. Analyzing what those clusters share, there are evidences that $\mathscr{C}_{\mathrm{PTC}\,5}$, $\mathscr{C}_{\mathrm{PTC}\,7}$, and $\mathscr{C}_{\mathrm{PTC}\,9}$ share an important symbol e in the distribution of symbols throughout the year, and $\mathscr{C}_{\mathrm{PTC}\,1}$ and $\mathscr{C}_{\mathrm{PTC}\,11}$ share the symbol f as predominant (Figure 6.11). For those cases, cluster-based models do not reveal an incremental forecasting skill, and the possible solution is to keep the simpler option, the one-size-fits-all model or a new cluster that aggregates those.

6.7 Conclusions

This chapter explores a method to achieve cluster-based load forecasting. Having fewer models to train and maintain, more data available to train and test within each cluster, and not needing to have a long set of historical data to train an individual model are relevant aspects of applicability of the model. However, individual models are expected to achieve better accuracy except when a large event changes the shape of the power load curve on specific assets.

As a first step, the method addresses the challenge of creating a dictionary of symbols that reflects the daily shapes and patterns of the power load curves. Snippet extraction and posterior clustering implement this first step, followed by an interpretation of the results. Then, the daily load curves are discretized into symbol sequences representing a summary of time series in a more reduced dimension. Thus, clustering techniques, such as k-medoids, are used to cluster the symbol sequences into groups, followed by an interpretation of each cluster. Those clusters are, therefore, the basis for training the model specifically for each cluster data sample.

When comparing results of one-size-fits-all model and cluster-based model, generally the cluster-based models are more accurate than one global model for PTD and/or PTC assets.

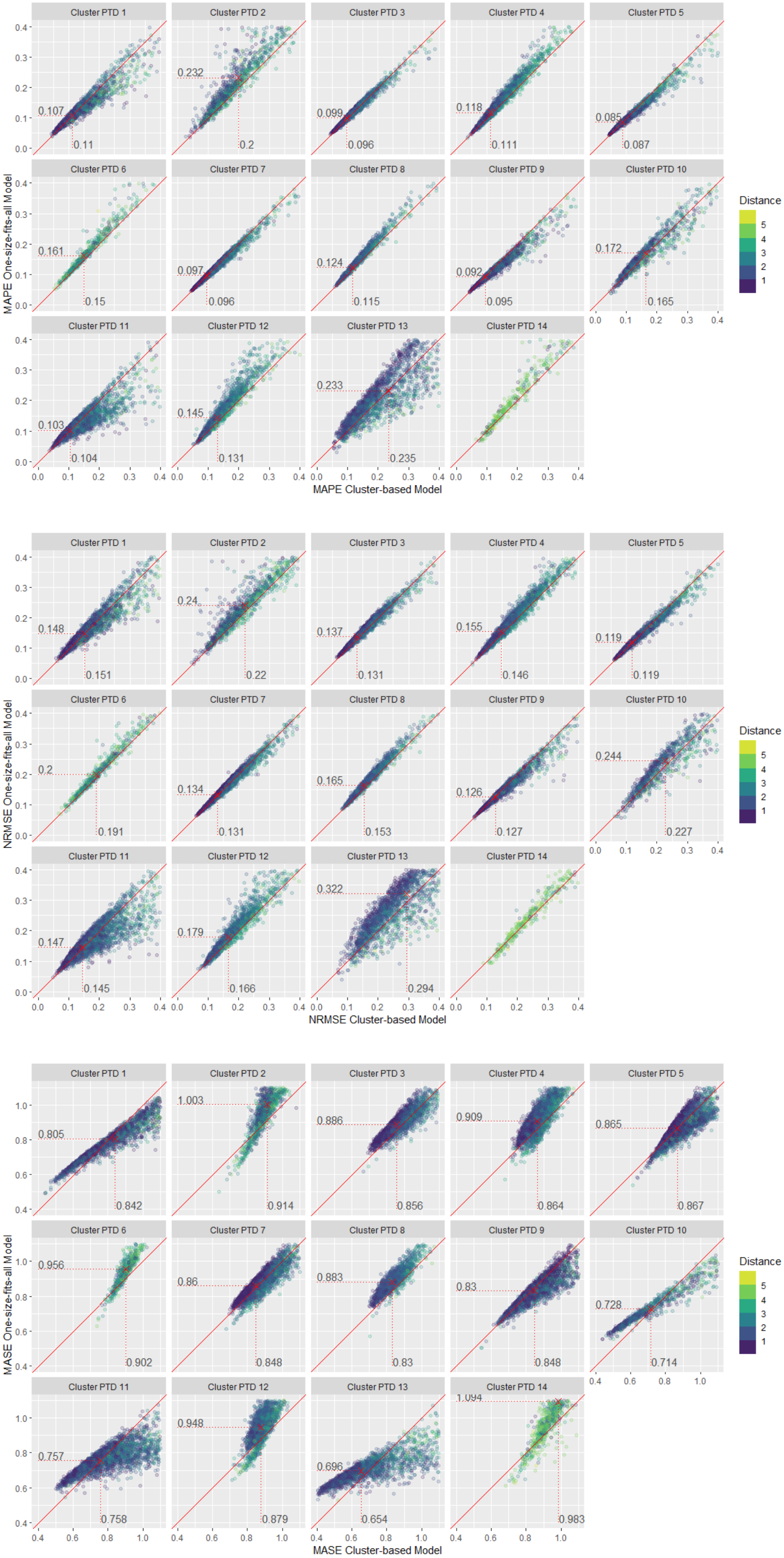


Figure 6.15. PTD scatter plot of MAPE, NRMSE, and MASE errors for each cluster-based model and the one-size-fits-all model. Each point is the forecasting error of a specific PTD time series colored by distance to the respective medoid. The points above the red line means that the cluster-based model is better than the other latter. The cross mark is the median center of cloud of points.

Chapter 7

Distributed System Architecture

7.1 Introduction

The disaggregated forecast problem challenges the capacity of storage and computing to deal with hundreds of thousands of time series and artifacts, such as fitted models, feature data tables, images, metrics, and logs. The system architecture was designed considering important requirements for a feasible system that would be integrated into existing DSO business processes and promote the continuous innovation process in energy management.

- **Big Data Storage** – There are three types of time series: (i) measurements from assets, (ii) meteorological observations and/or predictions, and (iii) predictions from the forecast procedure. In this scenario, each of 100 000 assets takes six different measurements[1] in each quarter hour. Six variables of numerical weather prediction (NWP) are stored, for 2257 Portugal locations with a resolution of 3 hours. Plus 100 000 time series with a resolution of 30 minutes as a result of the forecast. Without considering data structures, indexes, or overlap of time series predictions (that exists due to continuous window scrolling), these raw time series take up 170 GB[2] per year. Storage also needs to cope daily with new data appended to existing time series. Additionally, there are artifacts resulting from a session of model calibration that are also saved, such as the fitted model itself, images, metrics, and other data results.

- **Computing Scalability** – Individual model fitting for each asset implies computing and memory resources available to run the calibration, testing, and other artifact production in parallel. The same happens for the prediction, in this case with a higher priority to take the predictions in a useful time (a few hours to complete the forecast once the new data have been appended to time series). In addition, there are applied algorithms that, even parallelizable, involve the exchange of messages or other forms of synchronization among partial data executors.

- **Responsibility Segregation** – In a complex machine learning problem, different roles may be segregated, and each may use different tools and programming languages. Three roles were considered: (i) *data pipeline engineering* to continuously ensure that time series and metadata are extracted, transformed, and loaded from meteorological services and DSO systems into the big data storage; (ii) *data science* to address data discovery, data modeling, and model structure design to achieve a good performant methodology;

[1] Yet, a significant amount of assets only have 3 different measurements

[2] Time series elements as numerical values encoded by 8-bytes double.

(iii) *machine learning engineering* to ensure that the methodology designed by the data science role can work in a parallel computing scenario and be independent of the choices for data storage design. Considering that part of the disaggregated load forecasting was achieved by individual models, the data science role actually does not need to address whether the modeling code would run in parallel for 100 000 time series in a computing cluster or launch locally for a small set of time series. Indeed, the responsibility of the machine learning engineer is to prepare the environment including resources and input data, monitor and call the modeling code prepared by the data science role, and save the output results as designed in the big data storage.

- **Traceability** – Model lineage is required to judge the models created and audit its output. It is important to track the input arguments including raw data or the data feature, metrics and images that may be created during the code run, the versioned code itself, and the resulting fitted model. Additionally, inference outcomes to be auditable should be accompanied by the versioned fitted model.

- **Reproducibility** – The principle that the methodology applied is easily reproducible taking into account that in machine learning the pipeline code, input arguments, input data, and sometimes the seed used for algorithm random initializations are needed to get the same results in a later execution. In such an online system with a large number of time series continuously updated, it is relevant to make a decision on what data and how data are historically tracked for reproducibility purposes.

The following sections explain how these requirements were considered during system design and implementation. In fact, this resulted in an enterprise forecasting system called PREDIS – *PREvisão DIStribuída* whose outcomes have been used to anticipate load peaks and network constraints [21].

7.2 Software overview

The system is mainly developed using Java for data pipeline and machine learning (ML) engineering, R for data modeling and model calibration and inference (data science part) and also Scala for some additional tasks. The data pipeline and ML engineering part resulted in a set of modules:

- **PREDIS-API** module implements the data access object pattern according to the schema design for storage (see Section 7.3);

- **PREDIS-API-SPARK** module implements the interface between the DAO implementation and Spark nuances for ad-hoc queries or clustering tasks (see Section 7.4.2);

- **PREDIS-SERVER** and **PREDIS-CLIENT** modules implement a thrift-based server and client which offers services for application integration and launch forecasting and calibration procedures from orchestration software or manually;

- **PREDIS-DATA-INTEGRATION** module implements the data pipelines to extract, transform and load meteorological (FTP access) and power load (database access) time series and other metadata;

- **PREDIS-FORECAST** module has two parts: (i) the first part implements the R algorithms for calibration, testing and inference from a data science perspective, that is, the data modeling and the model structure, (ii) the second part implements the Java ML engineering to run the models in a parallel environment preparing the resources and input data, monitor and call the data science code, and finally save the output results as designed in the big data storage (see Sections 7.3.4 and 7.4.1);

- **PREDIS-WEBAPP** module implements (i) a web application with a map user interface to visualize geographically assets and its time series and metadata and (ii) a dashboard interface to visualize fitted models including metrics, images, and other resources resulting from fitting step, as exhibited in Figures 4.6 to 4.10.

  Daily forecasting is activated by orchestrating PREDIS-DATA-INTEGRATION during the early morning to load meteorological predictions and the last power load measurements of each asset, followed by the running of forecasting through PREDIS-CLIENT methods.

7.3 Storage

The big data storage used in the implementation is the HBase database deployed in a Hadoop ecosystem cluster with 15 region servers that serve data for read and write purposes, 2 master nodes that handle the region assignment and DDL operations, and 3 zookeeper nodes that offer a distributed coordination service for region assignments and recovery. The HBase is a random read/write access database capable of hosting very large tables atop clusters of commodity hardware. It is a distributed and scalable database whose design decisions chosen consistency and partition tolerance over availability from the CAP theorem perspective.

As a column-oriented database, the HBase data model conceptually follows a multidimensional map in which a value is stored considering the fully path: namespace, table, column family, column qualifier, cell, timestamp. A column in an HBase table consists of a column family and a column qualifier, which are generally delimited by a colon character. A cell is the intersection between this conceptual column and a row defined by a row key. Even a cell may have different values' versions, actually unequivocally identified by the timestamp written alongside each value. The actual physical view of how the tables are serialized to the disk dictated the best practices in schema design. To illustrate the point, let us give some examples: (i) the rows are primary indexed and sorted lexicographically by row key and major and minor compactions keep that while insert and delete operations are occurring, (ii) the tables are split over regions considering the row key as the value to set the interval of each region, (iii) it is the column family members (and not on the table level) that are stored together on the filesystem.

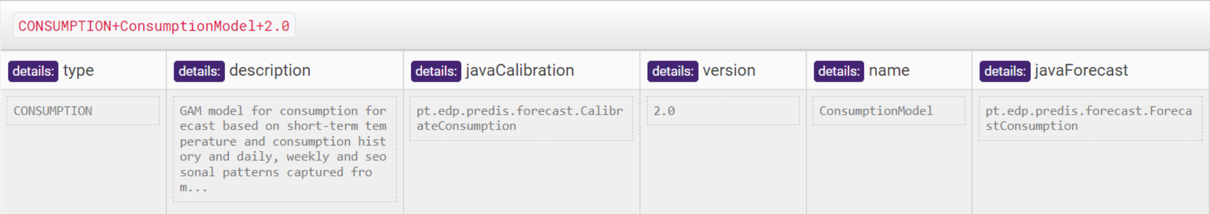
The data schema for the forecasting system consists of 6 main tables: ASSET, ASSET-GEOINDEX, MODEL-CONF, MEASUREMENT, FORECAST, and SESSION. The first three tables do not exhibit characteristics for the need of a big data storage; HBase was used to simplify the architecture and keep with a unique database instance. Indeed, the other three tables store a large amount of data volume.

7.3.1 Asset

The table ASSET maintains metadata related to the identification, localization, and characteristics of PTC and PTD power grid assets, where the row key is defined by the type plus the identification code of the asset, for example `PTD+1107D1016900`. The table ASSET-GEOINDEX is a projection of the same data but indexed, that is, the design of the row key, by localization using the geohash pattern.

7.3.2 Model configuration

The table MODEL-CONF consists of the configuration of the machine learning models that are being implemented. Alongside its identification by name and version, the configuration sets that Java classes are called during the calibration and inference phases, as shown in Figure 7.1, or as generically called the fitting and transform steps (see Figure 7.3).



CONSUMPTION+ConsumptionModel+2.0

details: type	details: description	details: javaCalibration	details: version	details: name	details: javaForecast
CONSUMPTION	GAM model for consumption forecast based on short-term temperature and consumption history and daily, weekly and seosonal patterns captured from...	pt.edp.predis.forecast.CalibrateConsumption	2.0	ConsumptionModel	pt.edp.predis.forecast.ForecastConsumption

Figure 7.1. An example row for table MODEL-CONF whose row key consists on type, name, and version of the model. These fields are also kept on respective columns besides the Java classes which are called during calibration and inference phases.

7.3.3 Time series

The tables MEASUREMENT and FORECAST use the same data scheme and are used to store time series data with small differences between them.

Both tables use the same column families (`details`, `tags`, `loadcurve`, and `timeseries_30D`), though differences might apply on the column qualifiers. The `details` and `tags` families store metadata related to the time series. The `loadcurve` and `timeseries_30D` families store the numeric data points, and the date and time to which the data point refers is modeled as the column qualifier (see Figure 7.2).

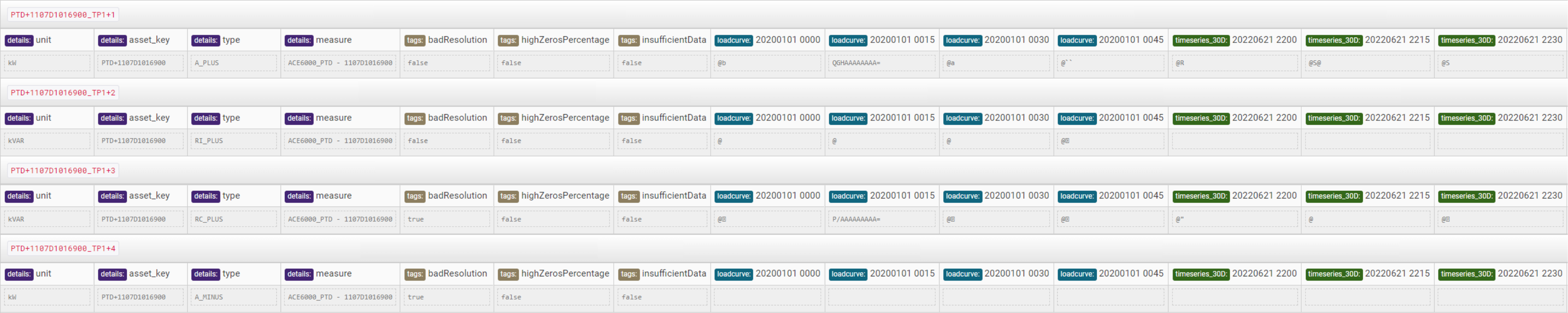
Note that the HBase allows for new column qualifiers at insert time. The same is not true for column families, which define the physical structure of the table. Physically, all column family members are stored together on the filesystem, and it is recommended that data be placed in the same column family if they have the same general access pattern. In this way, querying on a specific column family does not imply physically going over the data on other ones.

Considering that, the `loadcurve` family keeps the entire time series, which is needed during the calibration phase, for example. On the other hand, the `timeseries_30D` family keeps the exact same time series but trimmed to the last 30 days. This is faster when querying or scanning only the last few days, which is the most accessed part of time series, for example, during the inference phase or when serving predictions to another system. This trim is automatically achieved by the HBase feature that allows us to set a time-to-live (TTL) length that will automatically delete the cells, by internal minor compaction, once the expiration time is reached.

Note that there are also differences between these two tables.

In table MEASUREMENT, new data are mostly appended to existing time series. If updates occur on already stored values, this is due to data quality improvement and correction purposes, and the old versions of the value are not retained.

In table FORECAST, the range of predicted time series overlays part of time series already predicted in the previous inference run. For example, when the system predicts every day for the seven days, there is an overlay of the first six days as a sliding-window pattern. For



PTD+1107D1016900_TP1+1

details: unit	details: asset_key	details: type	details: measure	tags: badResolution	tags: highZerosPercentage	tags: insufficientData	loadcurve: 20200101 0000	loadcurve: 20200101 0015	loadcurve: 20200101 0030	loadcurve: 20200101 0045	timeseries_30D: 20220621 2200	timeseries_30D: 20220621 2215	timeseries_30D: 20220621 2230
kW	PTD+1107D1016900	A_PLUS	ACE6000_PTD - 1107D1016900	false	false	false	@b	QGHAAAAAAAA=	@a	@``	@R	@S@	@S

PTD+1107D1016900_TP1+2

details: unit	details: asset_key	details: type	details: measure	tags: badResolution	tags: highZerosPercentage	tags: insufficientData	loadcurve: 20200101 0000	loadcurve: 20200101 0015	loadcurve: 20200101 0030	loadcurve: 20200101 0045	timeseries_30D: 20220621 2200	timeseries_30D: 20220621 2215	timeseries_30D: 20220621 2230
kVAR	PTD+1107D1016900	RI_PLUS	ACE6000_PTD - 1107D1016900	false	false	false	@	@	@	@□			

PTD+1107D1016900_TP1+3

details: unit	details: asset_key	details: type	details: measure	tags: badResolution	tags: highZerosPercentage	tags: insufficientData	loadcurve: 20200101 0000	loadcurve: 20200101 0015	loadcurve: 20200101 0030	loadcurve: 20200101 0045	timeseries_30D: 20220621 2200	timeseries_30D: 20220621 2215	timeseries_30D: 20220621 2230
kVAR	PTD+1107D1016900	RC_PLUS	ACE6000_PTD - 1107D1016900	true	false	false	@□	P/AAAAAAAAA=	@□	@□	@"	@	@□

PTD+1107D1016900_TP1+4

details: unit	details: asset_key	details: type	details: measure	tags: badResolution	tags: highZerosPercentage	tags: insufficientData	loadcurve: 20200101 0000	loadcurve: 20200101 0015	loadcurve: 20200101 0030	loadcurve: 20200101 0045	timeseries_30D: 20220621 2200	timeseries_30D: 20220621 2215	timeseries_30D: 20220621 2230
kW	PTD+1107D1016900	A_MINUS	ACE6000_PTD - 1107D1016900	true	false	false							

Figure 7.2. An example row for table MEASUREMENT whose row key consists on asset type, asset code, and measurement id (see Table 4.1). The table consists on four main column families: `details`, `tags`, `loadcurve`, and `timeseries_30D`. In these last two column families, the date and time to which the data point refers is modeled as column qualifiers. Note that the numeric values are not exhibited.

traceability and audit purposes, it is important, in this case, to maintain the previous version of prediction values once the model outcomes may have been served to another system or business process. The timestamp concept is used here, and the table is configured to keep more than one version of the value. Therefore, even if a new value is stored in the exact same cell, the update operation keeps the previous value tagged by the timestamp.

7.3.4 Session Artifacts

The table SESSION maintains the artifacts resulting from the generic fitting step. Those artifacts such as the fitted model, metrics, images, logs, and other results related to the fitting process must be stored for traceability and evaluation purposes, and even to use the fitted model during the next step, which is generically called the transformation step. As shown in Figure 7.3 the fitting step can be the process of calibrating the forecasting model (as defined in Chapters 3 and 4), calibrating the ensemble model (as defined in Section 3.5), and finding the centroids during a clustering process (as defined in Chapter 6).

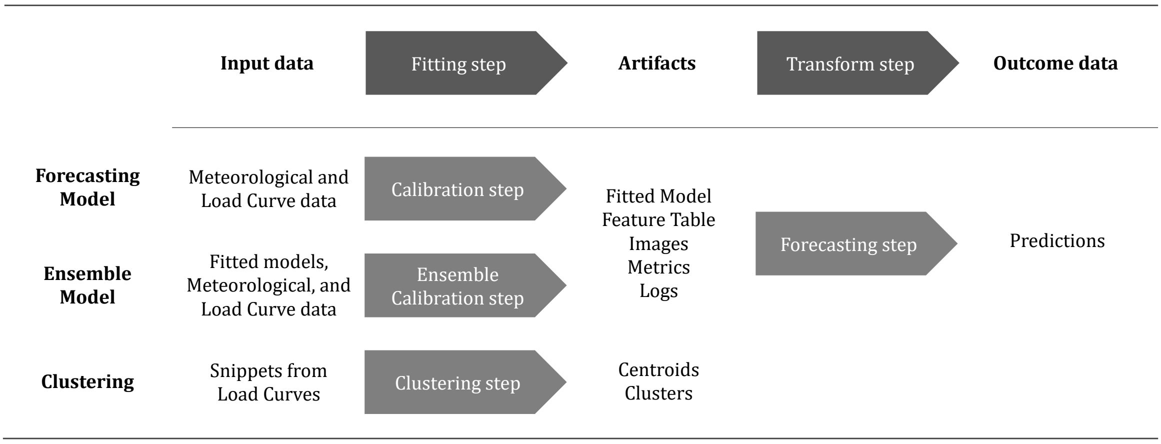


Figure 7.3. The fitting and transform step as generic steps. The fitting step outputs artifacts that must be stored for traceability and evaluation purposes and even to be used during the next step.

In the implementation, it is the data science code in the PREDIS-FORECAST module that is responsible for defining what should be traced (fitted models, feature tables, images, and metrics)[3]. Indeed, the main result of calling that data science code is essentially that trace declaration. The ML engineering code in the same module has the responsibility of storing all artifacts in a new row in the table SESSION. Note that HBase can technically handle binary objects within cells up to a default of 10MB, as the serialization of the fitted models and feature tables are considered medium size objects.

The row key is designed as the concatenation of measurement row key, model row key, and the date[4] that the step has been performed, for example `PTD+1107D1016900_TP1+1+CONSUMPTION+ConsumptionModel+2.0+20220805 1600`. So, it is possible to trace different versions of the model fitted with the same time series to compare them, or even trace the same model version calibrated in different dates if it is decided to refit the models when performance degrades or the time series drifts.

[3] In this implementation, logs are collected by the YARN log collector and are not stored in HBase.

[4] A UUID may be used to assure the uniqueness of row key, instead of the date/time the step has been performed.

7.4 Computing Engine

7.4.1 Individual Model Calibration and Forecasting

Such an important achievement of implementation is the handling of the calibration and forecasting of individual models, which means that each time series is individually used to fit the model and afterward to forecast. To address the challenge, the PREDIS-FORECAST module implements that ML engineering process to run in parallel in a scalable cluster.

As part of Hadoop ecosystem, YARN is the second generation of Hadoop's compute platform, whose new architecture decoupled the programming model from the resource management infrastructure [134]. The YARN architecture is based on two managers, which form the data-computation framework: resource manager (RM) and node manager (NM). The node manager is the per-machine daemon that is responsible for containers, monitoring their resource usage (CPU, memory, disk, and network) and reporting it to the resource manager/scheduler. The resource manager is the ultimate authority that arbitrates resources among all applications running in the system. It is composed of (i) the scheduler responsible for allocating resources to the various running applications subject to capacity constraints, resource availability, and user-allocation queues with diverse priorities, and (ii) the application manager (AMService) responsible for accepting job-submission, negotiating the first container for executing the application master and providing the services for restarting it in the case of failure. The Figure 7.4 shows this architecture and two applications running.

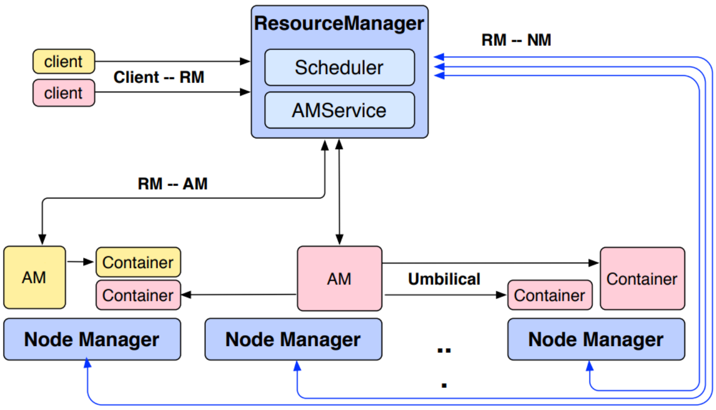


Figure 7.4. YARN architecture with the systems components in blue, and two applications running in yellow and pink. Figure based on [134].

In the implementation, the request to launch a calibration (or forecasting) job follows the sequence, as exhibited in Figures 7.5 and 7.6:

1. The `CalibrationService`, as a thrift server, receives calibration requests appointing the measurement and model keys and keeps these requests in a queue.
2. At a specific trigger time after the last calibration request received, `CalibrationService` starts the procedure to launch a new YARN application through the object `YARNClient`.
3. `YARNClient` is responsible for the preparation of a `submissionContext` object that contains all the information required by `ApplicationManager` to start the job. This includes
    - command for running the operating system process (in this case a JAVA application) and its initial arguments;

- local resources to be sent as the code (in this case the JAR file) and the list of commands sent through a file;
- shared resources to be sent to a distributed cache accessible by subsequent containers created by `ApplicationManager` (in this case a assembly file with JAVA and R code);
- security credentials and token renewer method if applicable (in this case, the access to HBase and the distributed cache must be guaranteed by these credentials);
- resource requirements for the `ApplicationManager` container as CPU and virtual cores;
- other application metadata as job name, type, priority queue request, and log aggregation configurations.

4. `YARNClient` submits the new job submission to `ApplicationManager` and waits until the job reaches the FINISH state (or FAIL or KILLED).

5. `ApplicationManager` and `Scheduler`, as components of `ResourceManager`, handle the negotiation of the first container, allocating the resources required according to the capacity constraints and the availability of the resources at the moment.

6. `ApplicationMaster` coordinates the logical plan of the calibration job by requesting resources from `ResourceManager`, generating a physical plan from the resources it receives, and coordinating the execution of the plan around faults. The `ResourceManager` remains ignorant of the semantics of each allocation.

7. `ApplicationMaster` implements a `AMRMClientAsync` that handles communication with the resource manager, with periodic heartbeats and status, and provides asynchronous updates on events to the `ApplicationMaster` logic, as `onContainersAllocated`, `onContainersCompleted`, `onNodesUpdated`,`onShutdownRequest`, and `onError`.

8. As requested containers are allocated, `ApplicationMaster` is notified and prepares the context for each container that provides command, security credentials, and local resources. This context is sent directly to each `NodeManager` who assigned the container to start the operating system process according to the context provided.

9. The container executes the command in parallel with other containers. In this case, they do not need to communicate between them, neither with `ApplicationMaster` until the end. The container receives a pair of keys that identify the measurement and the version of the model to be used. As the ML engineering part, the process communicates with HBase to obtain the meteorological and load curve time series and prepare the environment to call the data science code. Although the JAVA packages and the R code are obtained through the distributed cache, the R environment and the R packages are already installed[5] on each node running a `NodeManager`.

10. `ApplicationMaster` knows that the container is finished by communicating status messages with `ResourceManager`. The former has the responsibility to release the assigned container that has finished and retry if the container did not complete the task successfully.

[5] R environment and packages are prepared in advanced and distributed as a parcel using the feature of *Cloudera's Distribution for Hadoop* Management System. Native acceleration libraries such as OpenBLAS are available as system libraries previously installed on the operating system of each node.

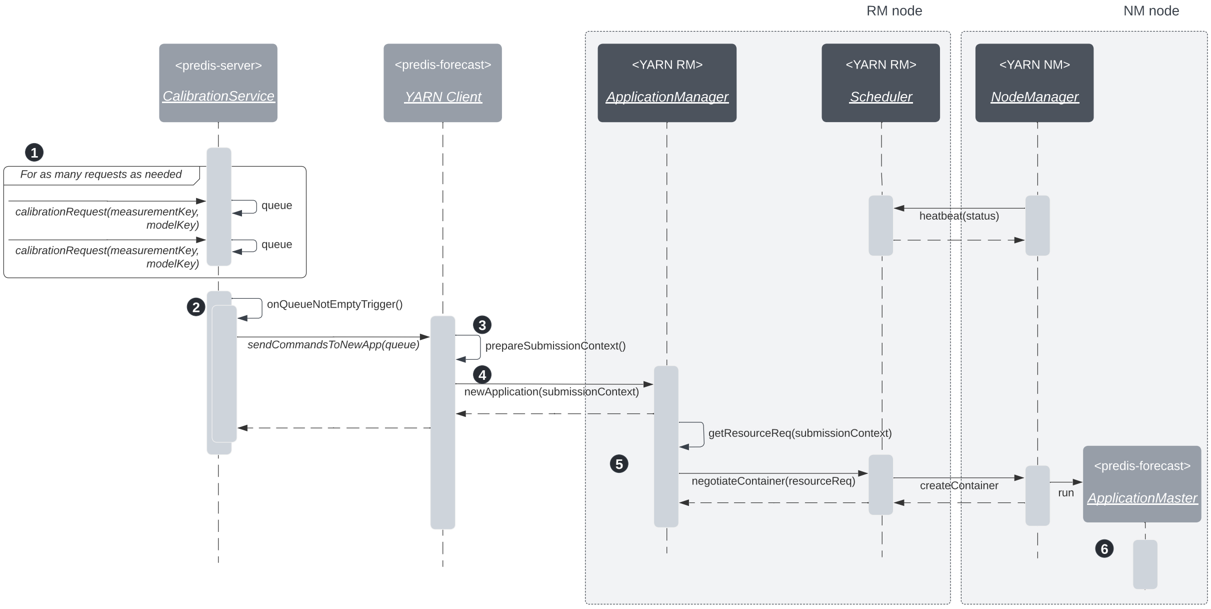

RM node
NM node
<predis-server>
CalibrationService
<predis-forecast>
YARN Client
<YARN RM>
ApplicationManager
<YARN RM>
Scheduler
<YARN NM>
NodeManager
1
For as many requests as needed
calibrationRequest(measurementKey, modelKey)
queue
calibrationRequest(measurementKey, modelKey)
queue
heatbeat(status)
2
onQueueNotEmptyTrigger()
3
sendCommandsToNewApp(queue)
prepareSubmissionContext()
4
newApplication(submissionContext)
getResourceReq(submissionContext)
5
<predis-forecast>
ApplicationMaster
negotiateContainer(resourceReq)
createContainer
run
6

Figure 7.5. The sequence of calls to launch the `ApplicationMaster` in a `NodeManager` node with available resources. `ApplicationManager` and `Scheduler` are responsible for accepting the job-submission asked by a `YARNClient` and negotiating the first container for executing the `ApplicationMaster`. It is the responsibility of the `ApplicationMaster` to negotiate more containers if needed and orchestrate the task.

7.4.2 Distributed Clustering

Clustering is a fundamental problem in data management and has a rich and notable history of publication of hundreds of different algorithms related to it. Nevertheless, a single method remains the most popular among the clustering methods, the k-means [135].

k-Means++ is a proposed version with a focus on obtaining a good initial set of centroids that is provably close to the optimum solution. k-Means|| explores how to scalable this initialization algorithm, obtaining a nearly optimal solution after a logarithmic number of passes. This initialization algorithm lends itself to a parallel implementation [136].

The Spark MLlib includes an implementation of the k-means|| algorithm. In Section 6.4.2, this library is used to address the clustering of snippets. Spark runs on the same YARN cluster as other applications as the application described in the previous Section 7.4.1.

7.5 Conclusions

The system is deployed in a Hadoop cluster with 22 servers. HBase, YARN, HDFS, and services related to Hadoop ecosystem are deployed in the same cluster, tweaking the CPU and memory shared by the services in accordance with Hadoop best practices.

The approaches described in previous sections are implemented resulting in a live daily forecasting system called PREDIS (Portuguese acronym for DIStributed PREdiction) whose results are used to anticipate load peaks and network constraints in the context of the Portugal distribution system. The system is recognized as an application prescribed and maintained by the DSO. This is possible by a distributed system architecture which copes with the challenge

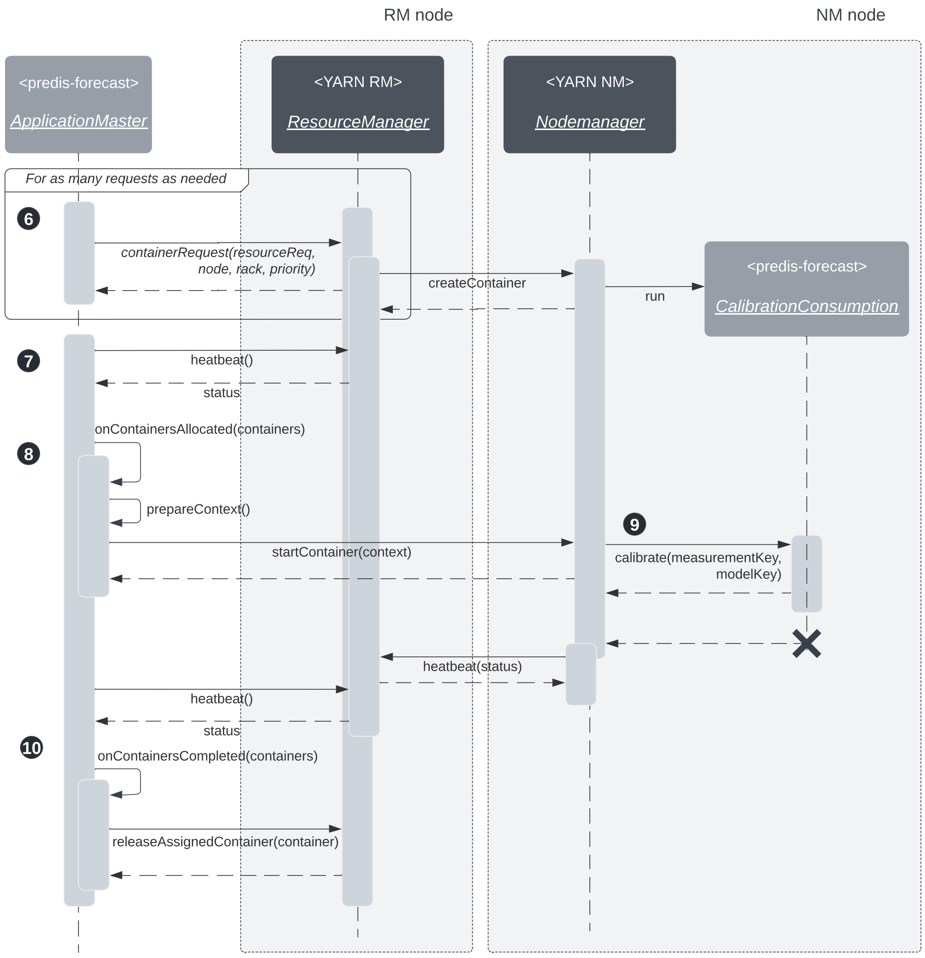


Figure 7.6. The sequence of calls to launch containers according to the logic plan coordinated by `ApplicationMaster`. As requested containers are allocated by the coordination between `ResourceManager` and `NodeManagers`, the coordinator of the application, the `ApplicationMaster`, sends the individual commands to fulfill its logic plan. The coordinator knows through asynchronous events when the containers are allocated, or completed without communicating directly to the containers.

of capacity and scalability for the storage and computing of hundreds of thousands of time series and artifacts.

One of the challenges is the applicability of individual forecasting models, which must be run daily to infer the next 24 hours. Specifically, the head end system for advanced metering infrastructure collects the updated power load time series throughout the day from all secondary substations and ends in the first hours of dawn. After updating the modeled data in Hbase, the system has a few hours to individually infer all 100 000 time series. Indeed, for the forecasting process of 100 000 individual models and considering 100 vcores available in the Hadoop system, the process takes 5h42’, which corresponds to 23d18h16’ vcores time, that is, the accumulated time the process would take sequentially. Note that the Hadoop cluster with 22 servers is configured up to approximately 550 vcores available to YARN in which the process runs. As the process scales linearly at least in the range of 100-550 vcores, the inference computing time is parallelizable enough to have the outputs in useful time to decision-marking and operations.

Chapter 8

Conclusions

The overall goal of this thesis is to improve the short-term load forecasting exercise. National load forecasting using statistical inference techniques was the first step in studying the structure of the model and the general explanatory variables that correlate with the power load. This step preceded the approach of the disaggregated power load curves at the secondary substation level, as a subject of forecasting. Individual models, cluster-based models, and one-size-fits-all models were explored as forecasting approaches within a spectrum of different accuracy and applicable effort to train and infer all secondary substations power load in useful time.

Short-term load forecasting at the low-voltage level, other than at the smart meter level, such as secondary substations, has not been as extensive. This thesis aims to contribute at that LV level with the introduced methodology and the data secured and collected by the DSO in the mainland of Portugal. Rather than tackle the few open datasets, a large dataset of power curves from all secondary substations in Portugal contributed to the study in a systematic way that incorporated different types and shapes of power consumption.

Each chapter had explored the steps and approach that contributed to the overall goal in ways which are analyzed in the following section.

8.1 Contributions

National Load Forecasting A classical regression model for system-level forecasting described by Tao Hong and his research group is used as a benchmark model. Following an additive approach to explanatory variables, the GAM technique provides a good balance between interpretability, ease of applicability, and accuracy with a diverse set of metrics to achieve an unbiased evaluation. A systematic approach improves the GAM-based model by introducing new synthetic explanatory variables based on the calendar, weather, and historical load. The resulting model is also compared to a gradient booster machine with the same explanatory variables after fair adjustments and hyperparameter optimization, which concludes that there are no improvements in terms of accuracy and a much poor level of interpretability. This chapter results in a foundational model structure and technique that will be explored at secondary substations.

Ensemble National Load Forecasting This section investigated the performance to ensemble different predictors to find a better major model. The ensemble learning technique used timely partitioned stacking, in which predictor weights were online found in different periods, such as seasons, weekends, August month, and public holidays periods like Christmas, New Year, Carnival, and Easter. This new ensemble method improves the final accuracy while still maintaining desirable interpretability.

Disaggregated Load Forecasting This chapter uses a new private dataset that encompasses all 100 000 secondary substations of the Portuguese power grid. In summary, this dataset contains 5 years of historical power load and numerical weather prediction in a spatial grid box. The same structure model with the nearest NWP of each asset is applied and fitted individually with each secondary substation time series. The error distribution of all forecasters are evaluated. All secondary substations are considered, even though the ones with very low and irrelevant power consumption, and thus with a higher error. This chapter results in the individual forecasters for each asset.

Power Load Classyfing using Shapelets This chapter explores the shapelets technique to capture time series patterns and curve shapes in order to cope with the consumption diversity. Among four use cases, a classifier is built to classify different types of power load of secondary substation (households, industries, services, utilities as water pumps or electrified transportation as electrified railroads). Shapelet technique creates interpretable classifiers and demonstrates the ability to extract interpretable patterns and knowledge from power load time series.

Power Load Clustering This chapter develops an appropriate data representation of power load time series, transforming them into discrete symbol sequences. A dictionary of symbols is kept that highlights and captures the daily shapes that occur in the power load curves within the year, week, and public holidays. Thus, the projection of one-year time series into a symbol sequence for all secondary substations are used to split the dataset into groups that contain the assets' load curve, which have similar daily shapes and patterns throughout the whole year.

Cluster-based Load Forecasting Cluster-based models are trained with a stratified sample of data from the respective cluster. The model structure and technique are similar. In addition, disaggregated load forecasting models of one-size-fits-all, are also fitted using a stratified sample of all PTC and PTD data regardless of their cluster. These two models are the baseline for comparison with the cluster-based and individual models. To pursue the goal of reducing the number of models to train and maintain, or when a new asset is deployed or no existing historical data are available to train the individual model but enough knowledge of what is or would be the general pattern, cluster-based models are a better solution than the one-size-fits-all GAM model. In general, cluster-based models achieve better accuracy than the one-size-fits-all model.

Distributed System Architecture The disaggregated forecast problem challenges the capacity of storage and computing to deal with hundreds of thousands of time series and artifacts, such as fitted models, feature data tables, images, metrics, and logs. The system architecture was designed considering important requirements, such as big data storage, computing scalability, responsibility segregation, traceability, and reproducibility, for a feasible system that would be integrated into existing DSO business processes and promote the continuous innovation process in energy management.

8.2 Further Work

The techniques explored in Chapter 6 can be combined using a different method to achieve STLF. The discretization of time series in Section 6.4 can be used to detect different regimes, which in Section 3.5 was based merely on strict calendar intervals. Once the regimes are no longer 100% calendar predictable, giving a power time series, a new classifier detects what the regime of the next day is. Although that classifier is not perfect, the method is applied when the classifier accuracy is superior enough (a threshold) to those assets with predictable next-day regime. Once again, the classifier could be trained globally, cluster-based, or individually per asset, and with different strategies, for example by historical subsequence search [122].

By splitting the power time series into regimes, different forecasters can be trained specifically for those regimes and joined into an ensemble model. In this case, each forecaster is trained for each symbol in the dictionary of Section 6.4.2, among others. Moreover, if using the fuzzy method to the classifier that detects the regime of the next, the ensemble model can consider, as ensemble weights, the degree of regime membership of the next day.

The other alternative is to explore the feature extraction applied to the residuals of the individual models instead of the raw time series. The residuals are usually scattered, not exhibiting any pattern, which indicates a potential good fit, but there are residuals showing no white noise, which indicates that the model structure or the fitting process was not good enough to capture the behavior of that stochastic process. Taking this into account, the goal of clustering residuals is to split assets whose stochastic process is not explained or captured entirely by the original model structure, with the prospect of designing new model structures or the fitting process among resulting clusters. In this case, a new dictionary of symbols must be computed following the same method for residuals as data.

In the domain of secondary substation load forecasting, boosting ensemble models can be effectively utilized in a sequential manner to achieve model individualization. The process begins with a one-size-fits-all model, and the resulting residuals from this model serve as inputs for cluster-based models. Subsequently, the individual model is employed. In this scenario, the complete chain, consisting of these three models, is employed for a subset of assets. However, for the remaining assets, which are comparatively easier to forecast, only a partial chain is required. An alternative approach is to initiate this chain of boosting ensembling with interpretable models, and then incorporate a second layer of models with lower interpretability but higher accuracy, depending on the specific needs.

References

[1] Jochen Markard. “The next phase of the energy transition and its implications for research and policy.” In: *Nature Energy* 3.8 (2018), pp. 628–633.

[2] Tao Hong. “Short Term Electric Load Forecasting.” PhD thesis. North Carolina State University, 2010.

[3] Adil Ahmed and Muhammad Khalid. “A review on the selected applications of forecasting models in renewable power systems.” In: *Renewable and Sustainable Energy Reviews* 100 (2019), pp. 9–21.

[4] Ahmad Nikoobakht et al. “Assessing increased flexibility of energy storage and demand response to accommodate a high penetration of renewable energy sources.” In: *IEEE Transactions on Sustainable Energy* 10.2 (2018), pp. 659–669.

[5] Luis Hernandez et al. “A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings.” In: *IEEE Communications Surveys Tutorials* 16.3 (2014), pp. 1460–1495.

[6] Sana Mujeeb, Nadeem Javaid, and Sakeena Javaid. “Data Analytics for Price Forecasting in Smart Grids: A Survey.” In: *2018 IEEE 21st International Multi-Topic Conference (INMIC)*. 2018, pp. 1–10.

[7] H Lee Willis. *Power distribution planning reference book, Second Edition, Revised and Expanded.* CRC press, 2004.

[8] Kasun Amarasinghe, Daniel L. Marino, and Milos Manic. “Deep neural networks for energy load forecasting.” In: *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. 2017, pp. 1483–1488.

[9] Hussein Jumma Jabir et al. “Impacts of Demand-Side Management on Electrical Power Systems: A Review.” In: *Energies* 11.5 (2018).

[10] João Peças Lopes et al. “Smart charging strategies for electric vehicles: Enhancing grid performance and maximizing the use of variable renewable energy resources.” In: *EVS24 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium* (2009).

[11] Alexandra M Oliveira, Rebecca R Beswick, and Yushan Yan. “A green hydrogen economy for a renewable energy society.” In: *Current Opinion in Chemical Engineering* 33 (2021), p. 100701.

[12] Benjamin F Hobbs et al. “Analysis of the value for unit commitment of improved load forecasts.” In: *IEEE Transactions on Power Systems* 14.4 (1999), pp. 1342–1348.

[13] Carol L Stimmel. *Big data analytics strategies for the smart grid.* Auerbach Publications, 2019.

[14] Grzegorz Dudek. “Pattern similarity-based methods for short-term load forecasting–Part 1: Principles.” In: *Applied Soft Computing* 37 (2015), pp. 277–287.

[15] Fotios Petropoulos et al. “Forecasting: theory and practice.” In: *International Journal of Forecasting* 38.3 (2022), pp. 705–871.

[16] Stephen Haben et al. “Review of low voltage load forecasting: methods, applications, and recommendations.” In: *Applied Energy* 304 (2021), p. 117798.

[17] Ran Li et al. “Development of low voltage network templates—Part I: Substation clustering and classification.” In: *IEEE Transactions on Power Systems* 30.6 (2014), pp. 3036–3044.

[18] Ran Li et al. “Development of low voltage network templates—Part II: Peak load estimation by clusterwise regression.” In: *IEEE Transactions on Power Systems* 30.6 (2014), pp. 3045–3052.

[19] Marco G. Pinheiro, Sara C. Madeira, and Alexandre P. Francisco. “Short-term electricity load forecasting—A systematic approach from system level to secondary substations.” In: *Applied Energy* 332 (2023), p. 120493.

[20] Marco G. Pinheiro, Sara C. Madeira, and Alexandre P. Francisco. “Shapelets to Classify Energy Demand Time Series.” In: *Energies* 15.8 (2022).

[21] Ricardo Gonçalves et al. “Forecasted chronological Power Flow for enabling timely dynamic tariff activation.” In: *CIRED 2019 Conference*. AIM, 2019.

[22] Ricardo Gonçalves et al. “PREDIS - State of the art cloud massive forecasting.” In: *IET Conference Proceedings* (2021), 1539–1541(2).

[23] Diogo Taborda et al. “Secondary substations smart metering campaign.” In: *CIRED-Open Access Proceedings Journal* 2017.1 (2017), pp. 2893–2896.

[24] Alireza Bahmanyar et al. “Emerging smart meters in electrical distribution systems: Opportunities and challenges.” In: *2016 24th Iranian Conference on Electrical Engineering (ICEE)*. IEEE. 2016, pp. 1082–1087.

[25] Niklas Löf et al. “Utilizing smart meters in LV network management.” In: *CIRED Seminar*. Vol. 2011. 2011, 21st.

[26] Jan G De Gooijer and Rob J Hyndman. “25 years of time series forecasting.” In: *International journal of forecasting* 22.3 (2006), pp. 443–473.

[27] Zhenyu Liu et al. “Forecast methods for time series data: a survey.” In: *IEEE Access* 9 (2021), pp. 91896–91912.

[28] P.D. Matthewman and H. Nicholson. “Techniques for load prediction in the electricity-supply industry.” In: *Proceedings of the Institution of Electrical Engineers* 115.10 (1968), p. 1451.

[29] Isaac Kofi Nti et al. “Electricity load forecasting: a systematic review.” In: *Journal of Electrical Systems and Information Technology* 7.1 (2020), pp. 1–19.

[30] Tao Hong et al. “Energy Forecasting: A Review and Outlook.” In: *IEEE Open Access Journal of Power and Energy* 7 (2020), pp. 376–388.

[31] G. Gross and F. D. Galiana. “Short-term load forecasting.” In: *Proceedings of the IEEE* 75.12 (1987), pp. 1558–1573.

[32] Ibrahim Moghram and Saifur Rahman. “Analysis and evaluation of five short-term load forecasting techniques.” In: *IEEE Transactions on power systems* 4.4 (1989), pp. 1484–1491.

[33] Hesham K Alfares and Mohammad Nazeeruddin. “Electric load forecasting: literature survey and classification of methods.” In: *International journal of systems science* 33.1 (2002), pp. 23–34.

[34] Heiko Hahn, Silja Meyer-Nieberg, and Stefan Pickl. “Electric load forecasting methods: Tools for decision making.” In: *European Journal of Operational Research* 199.3 (2009), pp. 902–907.

[35] L. Suganthi and Anand A. Samuel. “Energy models for demand forecasting — A review.” In: *Renewable and Sustainable Energy Reviews* 16.2 (2012), pp. 1223–1240.

[36] Tao Hong and Shu Fan. “Probabilistic electric load forecasting: A tutorial review.” In: *International Journal of Forecasting* 32.3 (2016), pp. 914–938.

[37] Corentin Kuster, Yacine Rezgui, and Monjur Mourshed. “Electrical load forecasting models: A critical systematic review.” In: *Sustainable cities and society* 35 (2017), pp. 257–270.

[38] Seyedeh Narjes Fallah et al. “Computational intelligence on short-term load forecasting: A methodological overview.” In: *Energies* 12.3 (2019), p. 393.

[39] S. Sp. Pappas et al. “Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models.” In: *Energy* 33.9 (2008), pp. 1353–1360.

[40] Yi-Shian Lee and Lee-Ing Tong. “Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming.” In: *Knowledge-Based Systems* 24.1 (2011), pp. 66–72.

[41] Yacine Chakhchoukh, Patrick Panciatici, and Lamine Mili. “Electric load forecasting based on statistical robust methods.” In: *IEEE Transactions on Power Systems* 26.3 (2010), pp. 982–991.

[42] Caston Sigauke and Delson Chikobvu. “Peak electricity demand forecasting using time series regression models: An application to South African data.” In: *Journal of Statistics and Management Systems* 19.4 (2016), pp. 567–586.

[43] Eduardo Caro, Jesus Juan, and Javier Cara. “Periodically correlated models for short-term electricity load forecasting.” In: *Applied Mathematics and Computation* 364 (2020), p. 124642.

[44] Tao Hong, Pu Wang, and H Lee Willis. “A naive multiple linear regression benchmark for short term load forecasting.” In: *2011 IEEE Power and Energy Society General Meeting*. IEEE. 2011, pp. 1–6.

[45] Amandine Pierrot and Yannig Goude. “Short-term electricity load forecasting with generalized additive models.” In: *Proceedings of ISAP power* 2011 (2011).

[46] Haeran Cho et al. “Modelling and forecasting daily electricity load via curve linear regression.” In: *Modeling and Stochastic Learning for Forecasting in High Dimensions*. Springer, 2015, pp. 35–54.

[47] Umberto Amato et al. “Forecasting high resolution electricity demand data with additive models including smooth and jagged components.” In: *International Journal of Forecasting* 37.1 (2021), pp. 171–185.

[48] Moshoko Emily Lebotsa et al. "Short term electricity demand forecasting using partially linear additive quantile regression with an application to the unit commitment problem." In: *Applied Energy* 222 (2018), pp. 104–118.

[49] Marisa Reis, André Garcia, and Ricardo J Bessa. "A scalable load forecasting system for low voltage grids." In: *2017 IEEE Manchester PowerTech*. IEEE. 2017, pp. 1–6.

[50] João Viana, Ricardo J Bessa, and João Sousa. "Load forecasting benchmark for smart meter data." In: *2019 IEEE Milan PowerTech*. IEEE. 2019, pp. 1–6.

[51] Alireza Khotanzad et al. "ANNSTLF-a neural-network-based electric load forecasting system." In: *IEEE Transactions on Neural networks* 8.4 (1997), pp. 835–846.

[52] Sharad Kumar, Shashank Mishra, and Shashank Gupta. "Short term load forecasting using ANN and multiple linear regression." In: *2016 second international conference on computational intelligence & communication technology (cict)*. IEEE. 2016, pp. 184–186.

[53] Ahmed Shaharyar Khwaja et al. "Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting." In: *Electric Power Systems Research* 179 (2020), p. 106080.

[54] Hyungeun Choi, Seunghyoung Ryu, and Hongseok Kim. "Short-term load forecasting based on ResNet and LSTM." In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE. 2018, pp. 1–6.

[55] Salah Bouktif et al. "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches." In: *Energies* 11.7 (2018), p. 1636.

[56] Bryan Lim et al. "Temporal fusion transformers for interpretable multi-horizon time series forecasting." In: *International Journal of Forecasting* (2021).

[57] Bo-Juen Chen, Ming-Wei Chang, et al. "Load forecasting using support vector machines: A study on EUNITE competition 2001." In: *IEEE transactions on power systems* 19.4 (2004), pp. 1821–1830.

[58] Dongxiao Niu, Yongli Wang, and Desheng Dash Wu. "Power load forecasting using support vector machine and ant colony optimization." In: *Expert systems with Applications* 37.3 (2010), pp. 2531–2539.

[59] Abbas Khosravi et al. "Interval type-2 fuzzy logic systems for load forecasting: A comparative study." In: *IEEE Transactions on Power Systems* 27.3 (2012), pp. 1274–1282.

[60] Tao Hong and Pu Wang. "Fuzzy interaction regression for short term load forecasting." In: *Fuzzy optimization and decision making* 13.1 (2014), pp. 91–103.

[61] Manish Kumar Singla and Sikander Hans. "Load forecasting using fuzzy logic tool box." In: *Global Research and Development Journal for Engineering* 38 (2018), pp. 12–19.

[62] Andrey Bogomolov et al. "Energy consumption prediction using people dynamics derived from cellular network data." In: *EPJ Data Science* 5 (2016), pp. 1–15.

[63] Jianzhou Wang et al. "Combined modeling for electric load forecasting with adaptive particle swarm optimization." In: *Energy* 35.4 (2010), pp. 1671–1678.

[64] Song Qiang and Yang Pu. "Short-term power load forecasting based on support vector machine and particle swarm optimization." In: *Journal of Algorithms & Computational Technology* 13 (2018), p. 1748301818797061.

[65] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. "Neural networks for short-term load forecasting: A review and evaluation." In: *Power Systems, IEEE Transactions on* 16.1 (2001), pp. 44–55.

[66] Weicong Kong et al. "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network." In: *IEEE Transactions on Smart Grid* 10.1 (2019), pp. 841–851.

[67] Muhammad Qamar Raza and Abbas Khosravi. "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings." In: *Renewable and Sustainable Energy Reviews* 50 (2015), pp. 1352–1372.

[68] Xiangyu Kong et al. "Short-term electrical load forecasting based on error correction using dynamic mode decomposition." In: *Applied Energy* 261 (2020), p. 114368.

[69] Joao Gama and Pedro Pereira Rodrigues. "Stream-based electricity load forecast." In: *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer. 2007, pp. 446–453.

[70] Mladen Kezunovic et al. "Big data analytics for future electricity grids." In: *Electric Power Systems Research* 189 (2020), p. 106788.

[71] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability." In: *Advances in neural information processing systems* 29 (2016).

[72] Rob J Hyndman and Anne B Koehler. "Another look at measures of forecast accuracy." In: *International journal of forecasting* 22.4 (2006), pp. 679–688.

[73] Stephen Haben et al. "A new error measure for forecasts of household-level, high resolution electrical energy consumption." In: *International Journal of Forecasting* 30.2 (2014), pp. 246–256.

[74] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[75] Trevor J Hastie and Robert J Tibshirani. "Generalized additive models." In: *Statistical Science* (1986), pp. 297–310.

[76] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*. Vol. 43. CRC Press, 1990.

[77] Simon N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman & Hall, 2006.

[78] Simon N. Wood. *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. R package version 1.8-38. 2021.

[79] Alexey Natekin and Alois Knoll. "Gradient boosting machines, a tutorial." In: *Frontiers in neurorobotics* 7 (2013), p. 21.

[80] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. "Additive logistic regression: a statistical view of boosting." In: *The annals of statistics* 28.2 (2000), pp. 337–407.

[81] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine." In: *Annals of statistics* (2001), pp. 1189–1232.

[82] Raza Abid Abbasi et al. "Short term load forecasting using XGBoost." In: *Workshops of the International Conference on Advanced Information Networking and Applications*. Springer. 2019, pp. 1120–1131.

[83] Xiaoqun Liao et al. "Research on short-term load forecasting using XGBoost based on similar days." In: *2019 International conference on intelligent transportation, big data & smart city (ICITBS)*. IEEE. 2019, pp. 675–678.

[84] Yuanyuan Wang et al. "Short-term load forecasting of industrial customers based on SVMD and XGBoost." In: *International Journal of Electrical Power & Energy Systems* 129 (2021), p. 106830.

[85] David Gunning et al. "XAI—Explainable artificial intelligence." In: *Science robotics* 4.37 (2019), eaay7120.

[86] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. "Explainable artificial intelligence: A survey." In: *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE. 2018, pp. 0210–0215.

[87] Plamen P Angelov et al. "Explainable artificial intelligence: an analytical review." In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11.5 (2021), e1424.

[88] Christoph Molnar et al. "General pitfalls of model-agnostic interpretation methods for machine learning models." In: *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*. Springer. 2022, pp. 39–68.

[89] Nick Littlestone and Manfred K Warmuth. "The weighted majority algorithm." In: *Information and computation* 108.2 (1994), pp. 212–261.

[90] Tom Terlouw et al. "Multi-objective optimization of energy arbitrage in community energy storage systems using different battery technologies." In: *Applied energy* 239 (2019), pp. 356–372.

[91] Wolf-Peter Schill. "Electricity Storage and the Renewable Energy Transition." In: *Joule* 4.10 (2020), pp. 2059–2064.

[92] Julio A Sanguesa et al. "A review on electric vehicles: Technologies and challenges." In: *Smart Cities* 4.1 (2021), pp. 372–404.

[93] Illia Diahovchenko et al. "Progress and challenges in smart grids: distributed generation, smart metering, energy storage and smart loads." In: *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 44.4 (2020), pp. 1319–1333.

[94] Kyriaki Psara et al. "European Energy Regulatory, Socioeconomic, and Organizational Aspects: An Analysis of Barriers Related to Data-Driven Services across Electricity Sectors." In: *Energies* 15.6 (2022).

[95] Bryan Lim and Stefan Zohren. "Time-series forecasting with deep learning: a survey." In: *Philosophical Transactions of the Royal Society A* 379.2194 (2021), p. 20200209.

[96] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering–a decade review." In: *Information Systems* 53 (2015), pp. 16–38.

[97] Hassan Ismail Fawaz et al. "Deep learning for time series classification: a review." In: *Data mining and knowledge discovery* 33.4 (2019), pp. 917–963.

[98] Mustafa Gokce Baydogan and George Runger. "Learning a symbolic representation for multivariate time series classification." In: *Data Mining and Knowledge Discovery* 29.2 (2015), pp. 400–422.

[99] Chin-Chia Michael Yeh et al. "Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile." In: *Data Mining and Knowledge Discovery* 32.1 (2018), pp. 83–123.

[100] Yan Zhu et al. "Matrix profile VII: Time series chains: A new primitive for time series data mining." In: *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2017, pp. 695–704.

[101] Shima Imani et al. "Introducing time series snippets: a new primitive for summarizing long time series." In: *Data Mining and Knowledge Discovery* 34.6 (2020), pp. 1713–1743.

[102] Shaghayegh Gharghabi et al. "Domain agnostic online semantic segmentation for multi-dimensional time series." In: *Data mining and knowledge discovery* 33.1 (2019), pp. 96–130.

[103] Md. Mehedi Hasan, Dhiman Chowdhury, and Md. Ziaur Rahman Khan. "Non-Intrusive Load Monitoring Using Current Shapelets." In: *Applied Sciences* 9.24 (2019).

[104] Bogdan-Petru Butunoi and Marc Frincu. "Shapelet based classification of customer consumption patterns." In: *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE. 2017, pp. 1–6.

[105] Fan Zhang, Chris Bales, and Hasan Fleyeh. "Feature Augmentation of Classifiers Using Learning Time Series Shapelets Transformation for Night Setback Classification of District Heating Substations." In: *Advances in Civil Engineering* 2021 (2021).

[106] Lipeng Zhu, Chao Lu, and Yuanzhang Sun. "Time series shapelet classification based online short-term voltage stability assessment." In: *IEEE Transactions on Power Systems* 31.2 (2015), pp. 1430–1439.

[107] Lipeng Zhu et al. "Imbalance learning machine-based power system short-term voltage stability assessment." In: *IEEE Transactions on Industrial Informatics* 13.5 (2017), pp. 2533–2543.

[108] Lipeng Zhu and David J Hill. "Networked Time Series Shapelet Learning for Power System Transient Stability Assessment." In: *IEEE Transactions on Power Systems* 37.1 (2021), pp. 416–428.

[109] Jesin Zakaria, Abdullah Mueen, and Eamonn Keogh. "Clustering time series using unsupervised-shapelets." In: *2012 IEEE 12th International Conference on Data Mining*. IEEE. 2012, pp. 785–794.

[110] Lexiang Ye and Eamonn Keogh. "Time series shapelets: a new primitive for data mining." In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 947–956.

[111] Abdullah Mueen, Eamonn Keogh, and Neal Young. "Logical-shapelets: an expressive primitive for time series classification." In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 1154–1162.

[112] Thanawin Rakthanmanon and Eamonn Keogh. "Fast shapelets: A scalable algorithm for discovering time series shapelets." In: *Proceedings of the thirteenth SIAM conference on data mining (SDM)*. SIAM. 2013, pp. 668–676.

[113] Josif Grabocka, Martin Wistuba, and Lars Schmidt-Thieme. "Fast classification of univariate and multivariate time series through shapelet discovery." In: *Knowledge and Information Systems* (2015), pp. 1–26.

[114] Ziqiang Cheng et al. “Time2graph: Revisiting time series modeling with dynamic shapelets.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 2020, pp. 3617–3624.

[115] Jon Hills et al. “Classification of time series by shapelet transformation.” In: *Data Mining and Knowledge Discovery* 28.4 (2014), pp. 851–881.

[116] Jason Lines and Anthony Bagnall. “Alternative quality measures for time series shapelets.” In: *Intelligent Data Engineering and Automated Learning-IDEAL 2012*. Springer, 2012, pp. 475–483.

[117] T Warren Liao. “Clustering of time series data—a survey.” In: *Pattern recognition* 38.11 (2005), pp. 1857–1874.

[118] A.B. Geva. “Non-stationary time-series prediction using fuzzy clustering.” In: *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397)*. 1999, pp. 413–417.

[119] E. Keogh, J. Lin, and W. Truppel. “Clustering of time series subsequences is meaningless: implications for previous and future research.” In: *Third IEEE International Conference on Data Mining*. 2003, pp. 115–122.

[120] Mohamed Chaouch. “Clustering-Based Improvement of Nonparametric Functional Time Series Forecasting: Application to Intra-Day Household-Level Load Curves.” In: *IEEE Transactions on Smart Grid* 5.1 (2014), pp. 411–419.

[121] Oleg Valgaev and Friederich Kupzog. “Building power demand forecasting using k-nearest neighbors model-initial approach.” In: *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. IEEE. 2016, pp. 1055–1060.

[122] Francisco Martínez-Álvarez et al. “A Novel Hybrid Algorithm to Forecast Functional Time Series Based on Pattern Sequence Similarity with Application to Electricity Demand.” In: *Energies* 12.1 (2019).

[123] Fateme Fahiman et al. “Improving load forecasting based on deep learning and K-shape clustering.” In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 4134–4141.

[124] Tri Kurniawan Wijaya et al. “Cluster-based aggregate forecasting for residential electricity demand using smart meter data.” In: *2015 IEEE International Conference on Big Data (Big Data)*. 2015, pp. 879–887.

[125] Yun Lu, Tiankui Zhang, and Zhimin Zeng. “Adaptive weighted fuzzy clustering algorithm for load profiling of smart grid customers.” In: *2016 IEEE/CIC International Conference on Communications in China (ICCC)*. 2016, pp. 1–6.

[126] Abbas Shahzadeh, Abbas Khosravi, and Saeid Nahavandi. “Improving load forecast accuracy by clustering consumers using smart meter data.” In: *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015, pp. 1–7.

[127] Athanasios Sfetsos and C Siriopoulos. “Combinatorial time series forecasting based on clustering algorithms and neural networks.” In: *Neural computing & applications* 13.1 (2004), pp. 56–64.

[128] Athanasios Sfetsos and Costas Siriopoulos. “Time series forecasting with a hybrid clustering scheme and pattern recognition.” In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 34.3 (2004), pp. 399–405.

[129] Padhraic Smyth. “Clustering sequences with hidden Markov models.” In: *Advances in neural information processing systems* 9 (1996).

[130] Ignacio Benítez and José-Luis Díez. “Automated Detection of Electric Energy Consumption Load Profile Patterns.” In: *Energies* 15.6 (2022), p. 2176.

[131] Michail Vlachos, Dimitrios Gunopulos, and Gautam Das. “Indexing time-series under conditions of noise.” In: *Data mining in time series databases*. World Scientific, 2004, pp. 67–100.

[132] Fei Wang et al. “An analysis of the application of simplified silhouette to the evaluation of k-means clustering validity.” In: *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2017, pp. 291–305.

[133] Ivan Calero et al. “Duck-Curve Mitigation in Power Grids With High Penetration of PV Generation.” In: *IEEE Transactions on Smart Grid* 13.1 (2021), pp. 314–329.

[134] Vinod Kumar Vavilapalli et al. “Apache hadoop yarn: Yet another resource negotiator.” In: *Proceedings of the 4th annual Symposium on Cloud Computing*. 2013, pp. 1–16.

[135] Xindong Wu et al. “Top 10 algorithms in data mining.” In: *Knowledge and information systems* 14.1 (2008), pp. 1–37.

[136] Bahman Bahmani et al. “Scalable K-Means+.” In: *Proceedings of the VLDB Endowment* 5.7 (2012).