University of Pavia

Ph.D. School in Electronics, Computer Science and Electrical Engineering

Cycle XXXIII

Machine Learning methods for long and short term energy demand forecasting

Author:
Alessandro Incremona

Supervisor:
Prof. Giuseppe De Nicolao

2017-2020

UNIVERSITY OF PAVIA

Abstract

Faculty of Engineering
Ph.D. School in Electronics, Computer Science and Electrical Engineering

Doctor of Philosophy

Machine Learning methods for long and short term energy demand forecasting

by Alessandro Incremona

The thesis addresses the problems of long- and short- term electric load demand forecasting by using a mixed approach consisting of statistics and machine learning algorithms. The modelling of the multi-seasonal component of the Italian electric load is investigated by spectral analysis combined with machine learning. In particular, a frequency-domain version of the LASSO is developed in order to enforce sparsity in the parametrization and efficiently obtain the main harmonics of the multi-seasonal term. The corresponding model yields one-year ahead forecasts whose Mean Absolute Percentage Error (MAPE) has the same order of magnitude of the one-day ahead predictor currently used by the Italian Transmission System Operator. Again for the Italian case, two whole-day ahead predictors are designed. The former applies to normal days while the latter is specifically designed for the Easter week. Concerning normal days, a predictor is built that relies exclusively on the loads recorded in the previous days, without resorting to exogenous data such as weather forecasts. This approach is viable in view of the highly correlated nature of the demand series, provided that suitable regularization-based strategies are applied in order to reduce the degrees of freedom and hence the parameters variance. The obtained forecasts improve significantly on the Terna benchmark predictor. The Easter week predictor is based on a Gaussian process model, whose kernel, differently from standard choices, is statistically designed from historical data. Again, even without using temperatures, a definite improvement is achieved over the Terna predictions. In the last chapter of the thesis, aggregation and enhancement techniques are introduced in order to suitably combine the prediction of different experts. The results, obtained on German national load data, show that, even in the case of missing experts, the proposed strategies yield more accurate and robust predictions.

To my family

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Electricity is a vital part of everyday's life, its availability being a prerequisite of almost every human activity in the modern world. Within the electricity network, a balance must be maintained between the power produced and the power demanded in order to guarantee a stable electricity supply and avoid dangerous unbalances. This to prevent expensive damages and unfortunate events such as blackouts, which obviously represent huge inconveniences to the population. There are however physical limitations to the real-time generation of electricity: startup times of most of the generation plants, bottlenecks within the distribution grid, and the aleatoriness of renewable energy sources are just some of the main examples.

Therefore the national Transmission System Operator (TSO) resorts to forecasting algorithms to know in advance the demanded power so as to ask the producers to generate a secured amount of electricity and maintain the balance within the grid. The management of power systems calls for predictors having different horizons and it is usual to distinguish among long-term predictions and short-term ones. The former ones refer to future times ranging from some months to 20 years ahead and are relevant to the future generation and distribution planning, maintenance schedules, purchasing of generating, transmission and distribution equipment. Because of the long horizon, these predictions strongly rely on the estimation of repetitive patterns such as yearly, weekly and daily seasonalities as well as economic, social and demographic factors.

Short-term forecasts refer instead to future times ranging from a few minutes to one day ahead. Typically, it is required to predict the whole 24-hour profile of the electric load, based on historical data up to the previous day and exogenous covariates such as weather forecasts. This kind of predictions are crucial for efficient allocation of power generation and system security. They also have a crucial impact on the energy market as accurate forecasts result in reduction of costs and positive impact on the environment while inaccurate predictions might cause relevant financial losses.

This thesis addresses both long and short-term forecasts. For what concerns the long-term prediction, the Italian national demand is taken as case of study. In the thesis it is shown that the quarter-hourly Italian load demand, if adequately preprocessed, exhibits a high repeatability over the years. This justified a modelling approach that combines Fourier analysis and machine learning. In particular, a frequency domain version of the LASSO, aimed at enforcing sparsity within the power spectral density of the load, was designed. In particular the newly proposed LASSO-FFT, squeezed the computational complexity from $O\left(n^2\right)$ to $O\left(n\log(n)\right)$. The resulting model highlighted interactions between the different periodicities (yearly and weekly), usually neglected by the simple additive model typically adopted in the literature. One-year ahead predictive performances demonstrated that the multi-seasonal component is remarkably stable through the years and accounts for a significant fraction of the load.

The short-term prediction problem was addressed for two scenarios, namely normal days and the Easter Week, using again the Italian demand as case study. In particular, for the one-day ahead quarter-hourly prediction during normal days, a multipredictor approach was investigated. In order to prevent overfitting Tikhonov regularization methods and Radial Basis Functions were applied to smooth the surface formed by the predictors weights.

The prediction during the Easter Week was reformulated as the problem of tracking the departure of the demand series from the ‘typical’ Easter Week load profile. A Gaussian Process Regression was proposed that uses a kernel based on the autocovariance estimated from the historical series.

For both short-term prediction problems, the resulting forecasters, despite the simplicity and the absence of exogenous covariates, such as weather forecasts, competed well with the Italian TSO Terna benchmark predictor. Another important finding was that a further significant improvement is achieved if the predictions generated by the proposed models are averaged with the ones generated by the Terna forecaster. The improvement obtained by simple aggregation is explained by the uncorrelatedness between Terna prediction residuals and ours, a feature possibly explained by the novelty of the models underlying our new forecasters.

The last chapter of the thesis is dedicated to the issue of forecast aggregation. The easy access to large volumes of information, the improved communication technologies and the increasing automation of the processes, besides representing remarkable advantages, pose also new challenges for what concerns the effective management of data. Among the possible data streams, it is becoming common to have multiple predictions originated by external experts, a notable example being offered by the energy market where load forecasts at local/global level and short-/medium/long term are produced by multiple subjects [1].

In the forecasting community, the combination of multiple learning algorithms and models is commonly adopted in order to improve forecasts [2]. Ensemble learning methods such as bagging, boosting and stacking rely on training several base learners in such a way that the combination of their outputs leads to a reduction of the variance (bagging) or bias (boosting) (see [3], [4], [5], [6], [7], [8]). However, when predictions come from external sources, it may be impossible to access their generating models and even information on their structure could be missing. Nonetheless, improvements could still be obtained by suitable aggregation methods. In the literature, a variety of techniques have been explored, ranging from average-based ones to more sophisticated machine learning approaches [9], [10], [11], [12], [13], applied with success to different fields, such as finance, weather and load demand forecasts [14], [15], [16].

However, there are issues that, though commonly encountered in practice, have not yet been satisfactorily addressed. The available predictions may come from models trained on old data, with suboptimal choice or use of some features. If the forecast is externally generated it may be impossible to precisely diagnose these shortcomings and fix them. Nevertheless, one may not want to drop the expert altogether, given that its generating model could exploit some relevant features that would be otherwise unavailable. Under these biases, elementary aggregation methods help to reduce the variance of the individual predictions, but the potentialities of each expert are fatally underutilized.

This motivates the search for more sophisticated aggregation strategies. In particular, the one-day ahead prediction of the German load demand is considered as case of study and smart aggregation and enhancement schemes were investigated in order to better combine the predictions of twelve ‘experts’ and obtain new forecasters which, besides being more performing, are also more robust when some experts are occasionally missing.

1.1 Thesis Overview

The thesis is organized in eight chapters (including the Introduction) and two appendices.

Chapter 2: Load forecasting overview
This chapter gives a general overview about the different types of load forecasting, the main approaches used in literature, and the performance metrics adopted to evaluate predictors capabilities.

Chapter 3: Statistical and machine learning forecasting methods
A concise description of the statistical and machine learning approaches employed in the thesis is given in this chapter. In particular, the following topics are addressed: Fourier analysis, MLP and RBF artificial neural networks, Gaussian Processes, and shrinkage methods, including Tikhonov regularization, smoothing splines and the LASSO.

Chapter 4: Spectral characterization of the multi-seasonal component of the Italian electric load: a LASSO-FFT approach

In this chapter, the modeling of the multi-seasonal component of the national electric load is investigated. Differently from additive models that consider just the sum of daily, weekly and yearly periodic components, in order to account for possible interaction terms, a full parametrization in the frequency domain is considered. In the case of quarter-hourly data, almost 1 million parameters are needed to specify the model, which motivates the development of efficient learning techniques capable of enforcing sparsity in the parameter space. For this purpose, a Least Absolute Shrinkage and Selection Operator with Fast Fourier Transform (LASSO-FFT) algorithm is devised, having $O\left(n\log(n)\right)$ complexity. Applied to Italian load data, the LASSO-FFT algorithm yields one-year ahead forecasts whose Mean Absolute Percentage Error ($MAPE$), is close to one-day ahead predictors currently used by the Italian Transmission System Operator.

Chapter 5: Regularization methods for the short-term forecasting of the Italian electric load.

In this chapter, the one-day ahead forecasting of the Italian electric load demand is addressed by resorting to predictors relying exclusively on the serial correlation of the demand time-series. After a suitable preprocessing step consisting of logarithmic transformation and 7-day difference, six alternative techniques are explored for estimating tomorrow's 24-hour profile sampled every quarter-hour. Due to the quarter-hourly framework (that is, 96 values per day), each of the 96 linear predictors has 96 weights, forming a 96x96 matrix, that can be seen and displayed as a surface sampled on a square domain. In order to reduce the model complexity, different models of the surface were explored, including regularization and sparsity approaches. The normal days of three test years were used to validate the proposed methods. In particular, they were compared to the predictions of the Italian Transmission System Operator (TSO) Terna, achieving promising results in terms of quarter-hourly MAPE and MAE. For all the proposed models, the prediction residuals were weakly correlated with Terna's ones. This suggested that further improvement could be achieved by forecasts aggregation. In fact, the aggregated forecasts produced relevant drops in terms of quarter-hourly and daily MAPE, MAE and RMSE (up to 30%) over the three test years considered.

Chapter 6: Short-term forecasting of the load demand during the Easter Week

In this chapter, the problem of short-term prediction of the quarter-hourly electric load demand of Italy during the Easter Week is addressed by using a Gaussian Process (GP) approach to track the difference between the target Easter Week and an average Easter Week load profile. Differently from standard GP approaches that employ 'canonical' kernels, the proposed method uses as kernel the autocovariance of the stochastic process, estimated from historical load series starting from 1990. For the first day of the Easter Week, three long-term strategies and a short-term one, trained on normal days, have been applied, achieving substantial improvements with respect to Terna benchmark predictor. The uncorrelatedness of the prediction residuals of the proposed technique with those of the Terna forecaster motivated the use of an aggregation method that yielded a final decrease of more than 40% over Terna in the main error indexes.

Chapter 7: Aggregation of nonlinearly enhanced experts with application to electricity load forecasting

Combining the predictions of different base experts is a well known approach used to improve the accuracy of time series forecasts. Forecast aggregation is becoming crucial in many fields, including electricity forecasting, as Internet of Things and Cloud technology give access to larger numbers of sensor data, time series and predictions from external providers. In this context, it is not uncommon that the failure of some experts causes relevant drops in the performances of the aggregated forecast when classical techniques based on linear averaging are applied. This might be a symptom of suboptimality of the individual experts, that do not fully exploit important predictors, e.g. calendar features that play a major role in the electrical demand profiles. In this chapter, we therefore present two non-linear strategies to obtain aggregated forecasts, starting from the availability of a set of base experts and the knowledge of some relevant predictor variables. The first approach, called aggregation of enhanced experts (AEE), enhances each individual expert and then feeds the enhanced forecasts into classical linear aggregation techniques. In the second approach, called enhanced aggregation of experts (EAE), the expert forecasts are nonlinearly combined with the predictor variables through an Artificial Neural Network (ANN). The case of missing expert forecasts is also considered via a statistically-based imputation method. The short-term prediction of German electrical load is used as a case study. Twelve base experts are enhanced with respect to calendar features, i.e. daytime and weekday. Compared to state-of-the-art aggregation methods applied to the not-enhanced set of experts, the proposed approaches not only improve the accuracy of aggregated forecast (up to 25% reduction of MAPE and RMSE), but are also robust with respect to missing experts.

Chapter 8: Conclusions

In the final chapter, the main findings are summarized and discussed.

Appendix A: Italian special days

This appendix lists Italian special days, accounting for Winter, Summer, national holidays and Easter holidays from 1990 to 2019.

Appendix B: Italian electric load short-term prediction: performances in yearly seasons and daily phases

In this appendix the performances of the Italian short-term predictors developed in Chapter 5 are evaluated over different seasons of the year and phases of the day.

1.2 Collaborations and related pubblications

This thesis has been partially supported by the Italian Ministry for Research in the framework of the 2017 Program for Research Projects of National Interest (PRIN), Grant no. 2017YKXYXJ.

The material of Chapter 4 partially appears in:

- A. Incremona, G. De Nicolao. “Spectral Characterization of the Multi-Seasonal Component of the Italian Electric Load: A LASSO-FFT Approach.”, 2019 IEEE Control Systems Letters. (Also presented in the 58^{th} IEEE Conference on Decision and Control, Nice, France, December $11^{th} - 13^{th}$ 2019).

The material of Chapter 5 partially appears in:

- A. Incremona, G. De Nicolao. “Regularization methods for the short-term forecasting of the Italian electric load.”, SUBMITTED TO International Journal of Forecasting.

The material of Chapter 6 partially appears in:

- A. Incremona, G. De Nicolao. “Short-term forecasting of the Italian load demand during the Easter Week.”, SUBMITTED TO Neural Computing and Applications Journal.

The material of Chapter 7 was developed with the collaboration of IBM Research, Dublin, within the European project “Generalized Operational FLEXibility for Integrating Renewables in the Distribution Grid (GOFLEX)”, which has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731232.

- A. Incremona, G. De Nicolao, et al. “Aggregation of nonlinearly enhanced experts with application to electricity load forecasting.”, SUBMITTED TO Applied Soft Computing Journal.

2

Load forecasting: an overview

In the field of power production and distribution, the availability of accurate forecasts is crucial in order to ensure the equilibrium between electricity demand and production. It is very important for the national trasmission system operator to know in advance the electric load demand of a whole country. Indeed it is essential to maintain the balance within the national power grid, in order to guarantee a stable electricity supply to all the consumers and to keep the power system safe from dangerous instabilities that could lead to blackout phenomena, which represent a serious risk from a social and economical perspective. As a matter of fact, in an electricity distribution grid, power transformer and battery-to-grid sizing and modelling are crucial for an efficient power distribution. The design specification of transformers for the optimization of load losses versus no-load losses depends directly on the characteristics of the load profile that the transformer is expected to be subjected to. Moreover, with the liberalization of the energy markets, the electricity forecasts play a key role within the energy market bidding because accurate forecasts entail a more efficient use of energy, resulting in a reduction of costs and a positive impact on the environment, while inaccurate predictions might cause relevant financial losses.

The power industry requires predictors with different time horizons for various purposes: mid- and long-term predictions refer to future times ranging from some months to years ahead. They are relevant to the future generations and distribution planning, maintenance schedules, purchasing of generating, transmission and distribution equipment and they strongly rely on the presence of repetitive periodic patterns, such as weekly and yearly seasonalities, and exogenous factors which slowly affect the behaviour of the load demand, e.g. related to economy, such as the Gross Domestic Product (GDP) [17], but also to geographic conditions, demography, and intervention events, such as holidays or other exceptional occurrences. Figure 2.1 shows an example of load demand of four countries over the years 2017-2018: in all cases the presence of a yearly pattern is evident. A zoomed view of the same profiles, shown in Fig. 2.2, highlights

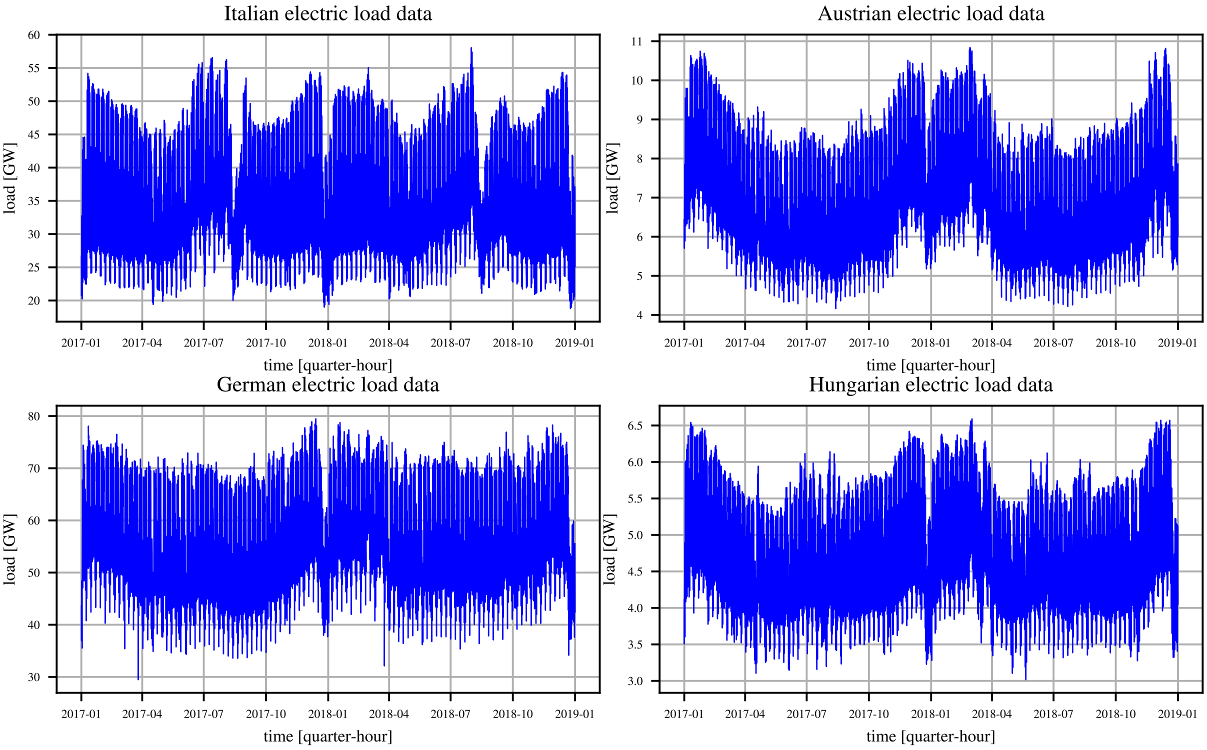


FIGURE 2.1: Load demand of Italy (top left), Austria (top right), Germany (bottom left) and Hungary (bottom right) over 2017 and 2018.

the weekly and daily periodicity of the signals. Finally in Fig. 2.3 it is possible to acknowledge how intervention events such as holidays can affect the typical seasonal patterns of the load demand time series.

In Fig. 2.4, the trend of the Italian electric load demand is compared with the Italian GDP per capita, after a suitable linear transformation:

$$\widehat{T} = \theta_1 + \theta_2 GDP$$

It is evident that there is a clear correlation between the two time series (e.g. the drop in correspondence of 2008, due to the economic crisis, affected both curves).

Short-term predictions refer instead to future times ranging from few minutes to one-day ahead. Typically, it is required to predict the whole 24-hour profile of the electric load, exploiting the strong correlation with previous data up to the day before and exogenous variables such as weather forecasts, which are covariates characterized by a high volatility and thus become relevant in a short-term scenario. These predictions are crucial for efficient allocation of power generation and system security and play a key role for maintaining the equilibrium between the generated and consumed power. Moreover, short-term load forecasts are of primary importance in the electricity market: as a matter of fact, market participants bid on the basis of the expected prices and quantities. If their expectations are correct, they can adjust their power

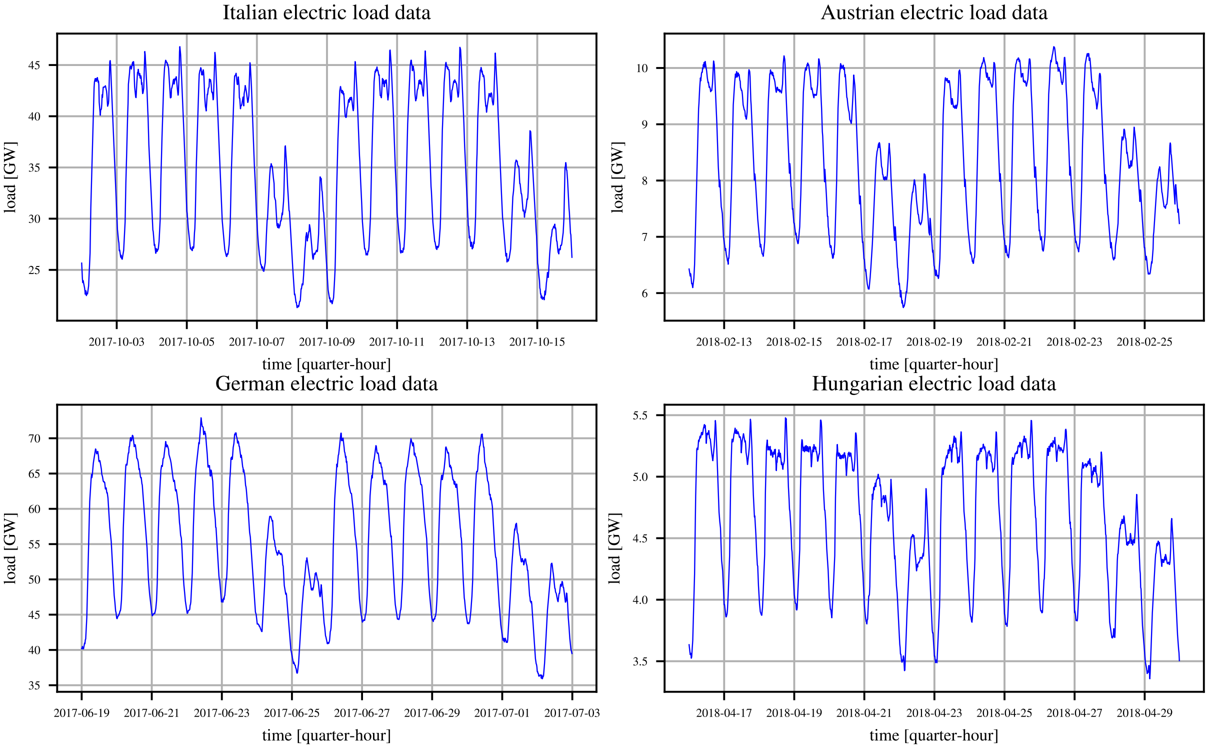


FIGURE 2.2: Load demand of Italy (top left), Austria (top right), Germany (bottom left) and Hungary (bottom right) over two weeks. There are clearly weekly and daily patterns whose shapes depend on the country and on the season.

plants accordingly and maximize their profits. Load forecasting problems can also be classified depending on the frequency of the observations (daily, hourly, half-hourly, quarter-hourly etc.). Typically, a finer sampling of the target variable implies a more difficult forecasting task and new variables come into play, such as daily periodicities in a quarter-hourly time series, that are not present in daily sampled ones.

2.1 Approaches and literature

Being an established research topic, electric load forecasting has a wide literature covering a variety of techniques: for what concerns classical statistical methods, some of the most popular techniques are time series decomposition, Exponential Smoothing (ES) [18], Autoregressive models (AR), Moving Average models (MA), Autoregressive Moving Average models [19], [20], [21] (with all their variants and extensions for including seasonal phenomena, exogenous variables and so on [22], [23]), non-parametric regression [24], semi-parametric regression [25], [26], state space models and Kalman filter [27], [28], [29]. In the recent years, machine learning and artificial intelligence techniques have been extensively applied to the energy forecasting field as well, with techniques such as Artificial Neural Networks (ANNs) [30], [31], [32], fuzzy logic [33], Support Vector Machines (SVMs) [34], [35], wavelets and statistical learning approaches

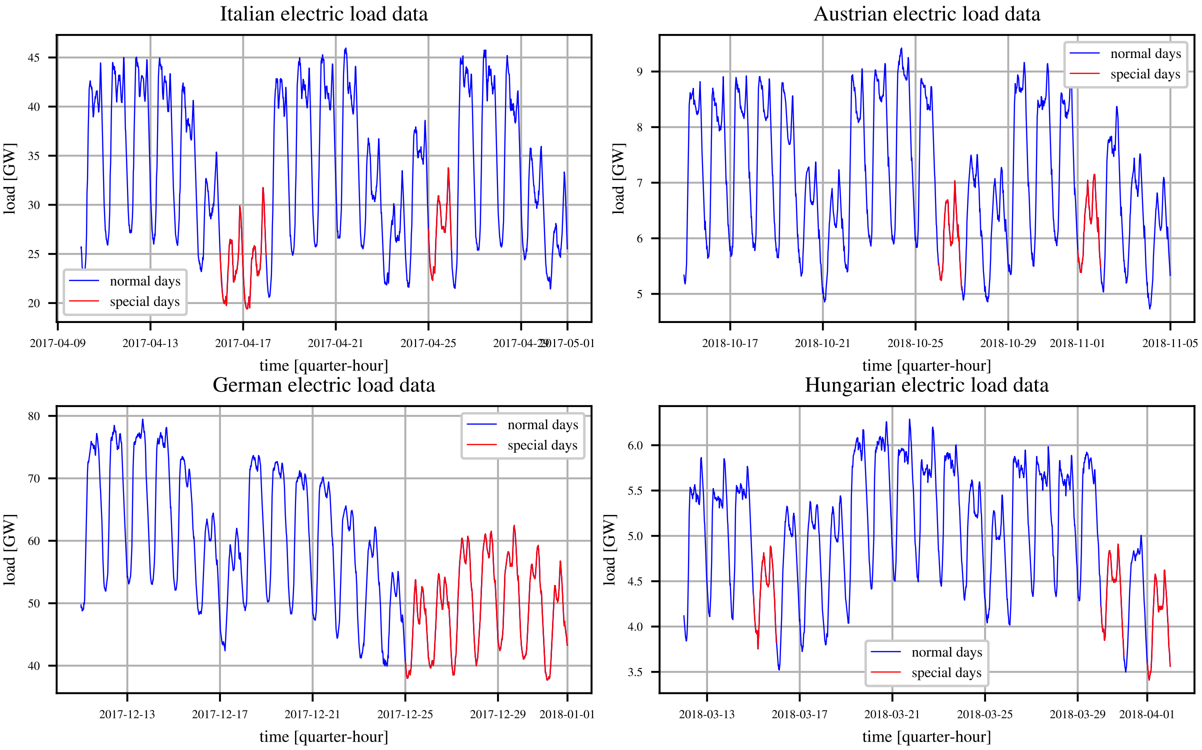


FIGURE 2.3: Load demand of Italy (top left), Austria (top right), Germany (bottom left) and Hungary (bottom right) over three weeks. For each country, a different season of the year is displayed, each one containing special days (highlighted in red). It is evident that, during special days, the typical seasonal pattern is blured, in some cases affecting also the normal days which are close to the holiday (this can be clearly seen in the case of the Christmas holidays of Germany, bottom left figure).

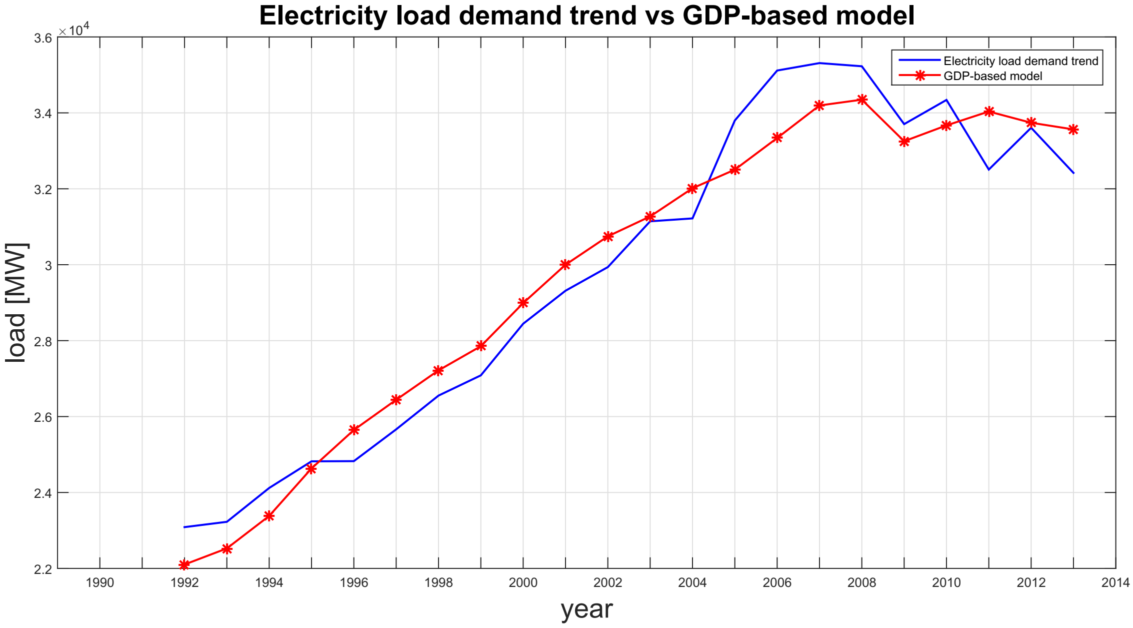


FIGURE 2.4: Italy: electric load demand trend vs GDP-based model (per capita).

(see [36], [37], [38], [39], [40], [41] and [42] for extensive reviews and surveys). Hybrid solutions have also been adopted recently, but several issues regarding training time, complexity, generalization, unclear structures and uninterpretable parameters remain for complex forecasting problems [43]. Since historical data may not be always available, qualitative methods have also been investigated (for example the Delphi method, Curve fitting and technological comparison with other methods [44]).

In general, there is no best model for load demand forecasting, but each technique can be a better or a worse solution according to several factors, such as the type of market, the market participants and the service area.

Statistical methods

Statistical methods include all those techniques based on the identification of load time series models using parameter estimation. Typical statistical approaches used in electric load forecasting are:

- Similar-day method
- Regression analysis
- Exponential smoothing (ES)
- Autoregressive Model (AR)
- Moving Average Model (MA)
- Autoregressive Moving Average Model (ARMA)
- Autoregressive Integrated Moving Average Model (ARIMA)
- ARX, ARMAX and ARIMAX models
- Seasonal Autoregressive Integrated Moving Average Model (SARIMA)

Similar-day method. This simple approach consists in forecasting the electric power load demand looking for a similar day in the historical data. The concept of similarity can be interpreted in terms of weather information, correspondence of weekday or day of the year. It is a technique that can be used for example for the so-called ‘special days’ such as holidays. For example to forecast the electric load demand of the 1 May 2018, which is a Tuesday, one could look for the electric load data of the most recent 1 May with the same weekday, that is 1 May 2012.

Regression analysis. This is a classical statistical technique used to estimate a regression function that can be used to predict the future power demand relying on the observed variables such as past demand, weather condition and day type.

Exponential Smoothing (ES). It consists of using a prediction which is based on past observations with an exponentially decaying weight:

$$\hat{l}_t = \alpha l_t + (1-\alpha)\hat{l}_{t-1} \tag{2.1}$$

with α, ranging from 0 to 1, being the exponential decay parameter, l the load demand observation and $\hat{l}$ the load demand prediction. The name 'Exponential Smoothing' derives from the fact that (2.1) can be rewritten as:

$$\hat{l}_t = \sum_{k=0}^{\infty} \alpha \left(1-\alpha\right)^k l_{t-k}$$

This method, also known as Holt-Winters predictor, may be too simple in order to adequately forecast the load demand, that is why a number of variants have been proposed in the literature. For instance, there are more flexible models allowing for a non polynomial trend and non constant seasonalities. There could also be parameters that can be tuned in order to minimize squared errors and adapt the model to the analysis objectives.

Autoregressive Model (AR). The electric load demand is modelled as a stochastic process whose current value is the linear combination of past values and a noise term $\epsilon \sim WN(0, \sigma^2)$.

$$L(t) = \sum_{i=1}^{p} \phi_i B^i L(t) + \epsilon(t)$$

This model is called Autoregressive (AR) model: p is the AR order, ϕ_i are the AR coefficients, typically obtained through least squares parameter estimation and B is the lag operator, defined as $B^i y(t) = y(t-i)$.

Moving Average Model (MA). The moving average model specifies that the current value of a stochastic process is given by the linear combination of the current and past values of a noise term.

$$L(t) = \sum_{i=1}^{q} \theta_i B^i + \epsilon(t)$$

This model is not very popular in load forecasting, since high-order MA models would be needed in order to reflect the serial correlation between current and past values of the electric load demand.

Autoregressive Moving Average Model (ARMA). This model represents a combination of the two previous techniques, since it represents the current value of a stationary process as the linear combination of its past values and current and previous values of a noise term.

$$L(t) = \sum_{i=1}^{p} \phi_i B^i L(t) + \epsilon(t) + \sum_{i=1}^{q} \theta_i B^i \epsilon(t)$$

The model order (p, q) can be choosen according to the Akaike criterion, while prediction error minimization can be used to estimate the coefficients [45]. The ARMA model is widely used in Short Term Load Forecasting (STLF).

Autoregressive Integrated Moving Average Model (ARIMA). The ARIMA model is an evolution of the ARMA model, characterized by the presence of an integral term, that is used in cases in which the time-series shows non stationarity:

$$\phi(B)\nabla^d L(t) = \theta(B)\epsilon(t)$$

where

$$\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$$

$$\theta(B) = 1 + \sum_{i=1}^{q} \theta_i B^i$$

$$\nabla^d = (1 - B)^d$$

In this expression we can see the presence of a difference operator of order d in addiction to the AR and MA part of the model.

ARX, ARMAX and ARIMAX models. They are the natural extensions of AR, ARMA and ARIMA models, used by some authors in order to take into account the presence of exogenous factors such as weather condition and economic events; as a matter of fact, the X stands for eXogenous. For instance, the ARMAX model can be written as follows:

$$\phi(B)L(t) = \theta(B)\epsilon(t) + \sum_{j=1}^{k} \psi^j(B) v^j(t)$$

where

$$\psi^j(B) = \sum_{i=1}^{r_j} \psi_i^j B^i$$

$v^j(t)$, $j = 1, \ldots, k$, denotes the $j - th$ exogenous variable and r_j is the order of the $j - th$ exogenous polynomial whose coefficients are ψ_i^j.

Seasonal Autoregressive Integrated Moving Average Model (SARIMA). This model is adopted for time-series showing both non stationarity and seasonality, which in the case of electric load demand is the weekly periodicity. The following is an example of multiplicative Seasonal ARIMA model:

$$\phi(B)\Phi(B^s)\nabla^d\nabla_s^D y(t) = \theta(L)\Theta(B^s)\epsilon(t)$$

where s is the weekly period, $\nabla_s = (1 - B^s)$ is the seasonal difference, d and D are the order of differencing and Θ and Φ are polynomial functions defined as follows:

$$\Phi(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{2s} - ... - \Phi_P B^{P_s}$$

$$\Theta(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{2s} + ... + \Theta_Q B^{Q_s}$$

Computational intelligence methods

They include a set of techniques that allow to compute load forecasts taking into account different factors such as previous loads and previous and current exogenous factors such as weather, social, economic conditions. The main advantages of these technique are related to the learning capabilities, massive parallelism, robustness, fault tolerance, while the main drawbacks are mainly related to the complexity in terms of time required, number of parameters to estimate, local minima and convergence. Some of the most popular computational intelligence techniques for load forecasting are:

- Artificial Neural Networks (ANNs)
- Fuzzy logic
- Support Vector Machines (SVMs)

Feedforward Artificial Neural Networks (ANNs). Neural networks are modular and flexible models used for black-box identification of non linear models. A feedforward neural network is structured as a network of fundamental units, called neurons, which are organized in three layers: input, hidden and output. Each neuron elaborates the input signals (which can come from other neurons) in order to obtain the desided output. They are characterized by an activation threshold, represented as an activation function, which can be either linear or nonlinear. As in the case of statistical methods, also for neural networks a careful choice of the explanatory variables and a model order selection phase are required. In the field of electric load forecasting, NNs are popular especially in STLF, because of their ability to address nonlinear modeling of large multivariate datasets.

Fuzzy logic. The goal of fuzzy logic is to aggregate data and create the rules that describe a certain system by using linguistic expressions. It is based on truth values that are real numbers ranging from 0 to 1. Different fuzzy logic approaches have been proposed over the years, mostly for STLF, also in combination with other techniques such as supervised learning [46], wavelet transforms and neural networks [47]. A neuro-fuzzy approach, where the tuning of the STLF parameters is performed by a Genetic Algorithm, has also been proposed [48].

Support Vector Machines (SVMs). It is a supervised learning model that defines a non-linear mapping of the data into a high dimensional space and it can be used for both classification and regression analysis. This approach has been applied recently in electric load forecasting problem (see [34] and [35]). It has also been used in combination with other techniques to obtain hybrid solutions suitable with the various facets of the load forecasting problem.

2.2 Accuracy measures

The accuracy of forecasts can be measured with a great variety of performance metrics, which can be classified, according to [49], in four categories:

- Scale-dependent
- Percentage
- Relative
- Scale-free

Scale-dependent

All the accuracy measures based on the forecast error $e = Y - F$, where we denote with Y the actual value and with F the forecast, belong to the scale-dependent category. Typical scale-dependent error measures are the Mean Absolute Error (MAE), the Geometric Mean Absolute Error ($GMAE$), the Mean Squared Error (MSE) and the Root Mean Squared Error ($RMSE$).

Mean Absolute Error. It is one of the most used metrics because of its simplicity and interpretability. The purpose of the absolute value is not to consider the error direction. It is defined as

$$MAE = \frac{\sum_{i=1}^{n} |Y_i - F_i|}{n}$$

Geometric Mean Absolute Error. It is defined as

$$GMAE = \left(\prod_{i=1}^{n} |Y_i - F_i| \right)^{\frac{1}{n}}$$

According to some authors, geometric error measures are useful when dealing with intermittent-demand data [50]. The con of this approach is that the $GMAE$ equals zero when at least one sample of the residual does.

Mean Squared Error. It is defined as the average of squared differences between the observations and the predictions. It is particularly useful to monitor the MSE of a prediction since it is very sensitive with respect to very large errors. Thanks to the presence of the square operator in place of the absolute value, not only the MSE does not consider the error direction, but it has desirable mathematical properties which make it the most commonly used cost function within optimization and predictive models training and calibration procedures. On the other hand, the main flaw of this metric is the interpretation, which is more complex with respect to the case of MAE since MSE has the same units of measurement as the square of the quantity being estimated.

$$MSE = \frac{\sum_{i=1}^{n} (Y_i - F_i)^2}{n}$$

Root Mean Squared Error. It is one of the most commonly used metrics for assessing accuracy. It is equal to the square root of the MSE and it has the advantage of being more interpretable because it has the same units as the observations.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (Y_i - F_i)^2}{n}}$$

Percentage

Scale-dependent error metrics can be problematic when it is necessary to compare performances over different time series. For such cases it can be more meaningful to consider the percentage error and its associated metrics such as the Mean Absolute Percentage Error ($MAPE$) and the symmetric Mean Absolute Percentage Error ($sMAPE$).

Mean Absolute Percentage Error. It is one of the most popular used performance index in the literature and in the industry because of its intuitive interpretation in terms of relative error.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{Y_i - F_i}{Y_i}\right|$$

However, being based on the percentage error, the main con of the $MAPE$ is that it becomes infinite if there are zero values in the series and its distribution is skewed when the actual value is close to zero.

Symmetric Mean Absolute Percentage Error. The $sMAPE$ was introduced by [51] to fix the asymmetry between positive and negative errors of the $MAPE$.

$$sMAPE = \frac{200}{n}\sum_{i=1}^{n}\frac{|Y_i - F_i|}{Y_i + F_i}$$

Relative

An alternative to percentage error metrics which allows to have a scale-independent metric is to consider the relative error, which is the ratio between the error of the forecast to be evaluated and the error obtained using a benchmark method $re = \frac{e}{e^*}$ [52], [53]. An usual choice for the benchmark method is the naïve forecast, where each $F_i = F_{i-1}$ or, in case of periodic time series with period T, the seasonal naïve forecast, where $F_i = F_{i-T}$. Of course the use of the naïve forecast as a benchmark is not possible if the target time series has an intermittent behaviour or is characterized by very small values. An example of performance metrics based on the relative error are the Median Relative Absolute Error (MdRAE) and the Geometric Mean Relative Absolute Error (GMRAE).

Median Relative Absolute Error. The use of the median makes this metric very robust against outliers. It is defined as

$$median(|re|)$$

Geometric Mean Relative Absolute Error. Its formulation is similar to the $GMAE$ but it is scale-independent because it considers the relative error instead of the forecast error.

$$GMRAE = \left(\prod_{i=1}^{n}|re_i|\right)^{\frac{1}{n}}$$

Scale-free

In order to overcome the drawbacks of the previously described methods, in recent years some authors proposed a new set of methods which are independent of the scale of the data. In the following the Mean Absolute Scaled Error (MASE) introduced by [49] is described.

Mean Absolute Scaled Error. The concept of this technique is to obtain a scaled error by dividing the errors by the in-sample MAE of the naïve forecast, that is the Mean Absolute Error obtained by the naïve forecast on the training set data:

$$se = \frac{e}{MAE_{in-sample}(e^*)}$$

Then, the Mean Absolute Scaled Error is given by

$$MASE = \frac{\sum_{i=1}^{n} |se|}{n}$$

The idea of using the in-sample MAE guarantees that the $MASE$ is always defined, even with intermittent time series data. The $MASE$ can be adopted to compare forecast methods on a single time series or on multiple ones.

In order to make comparison with the state-of-art techniques that can be found in the literature, the performance metrics that are adopted along this thesis are the $MAPE$, the MAE and the $RMSE$.

3

Statistical and Machine Learning forecasting methods

3.1 Fast Fourier Transform

Fourier analysis is a powerful tool in signal theory, since it simplifies signals analysis through the exploitation of trigonometric functions. In particular, switching to the frequency domain it is possible to examine the spectral density of a signal instead of its profile over time. This can be done through the Fourier transform:

$$\hat{s}(f) = \int_{-\infty}^{\infty} s(t) e^{-2\pi i t f}\, \mathrm{d}t.$$

The original signal $s(t)$ can be reconstructed from $\hat{s}(f)$ using the inverse Fourier transform:

$$s(t) = \int_{-\infty}^{\infty} \hat{s}(f) e^{2\pi i t f}\, \mathrm{d}f.$$

In the context of load forecasting, Fourier theory becomes particularly useful when dealing with a periodic time series. As a matter of fact it is possible to represent a periodic signal as the sum of properly weighted periodic signals (i.e. Fourier expansion), whose representation in the frequency domain is given by a discrete spectrum (shaped as a 'comb'). For discrete time series with length N, the Discrete Fourier Transform (DFT) and the Inverse Discrete Fourier Transform (IDFT) are considered:

$$\hat{s}(k) = \sum_{n=0}^{N-1} s(n) e^{-\frac{2\pi i}{N} kn}$$

$$s(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{s}(k) e^{\frac{2\pi i}{N} kn}$$

Thanks to the frequency domain representation, it is possible to detect the most relevant harmonic components and estimate the power spectral density of a signal from a sequence of time samples. Another advantage of working in the frequency domain is that the DFT (and IDFT) can be computed using the Fast Fourier Transform (FFT) (and Inverse Fast Fourier Transform (IFFT)), a fast algorithm that reduces the complexity of the operation from $O(n^2)$ to $O(nlog(n))$.

3.2 Multi-Layer Perceptron

Multi-Layer Perceptrons (MLP) belong to the category of feedforward Artificial Neural Networks (ANN). They are machine learning models characterized by a certain number of layers, each one with a certain number of neurons, which are the basic units, linked through weighted connections. The first layer is the input layer, where each neuron represent an input feature. The neurons in the hidden layers perform a weighted linear summation of the values coming from the previous layer and apply a nonlinear transformation before sending the output to the next layer. The nonlinear trasformation depends on the selected 'activation function'. There are different possible choices for the activation function, the most common being sigmoidal functions such as the hyperbolic tangent $tanh(x)$ and the logistic function $(1 + e^{-x})^{-1}$.

$$Y = w_{00} + \sum_{i=1}^{n_h} w_{i0} z_i$$

$$z_i = \gamma \left(\sum_{j=1}^{m} w_{ij} x_j + w_{0j} \right)$$

The above formulas refer to an MLP network with one hidden layer: Y is the output of the MLP, $x \in \mathbb{R}^m$ denotes the vector of input variables, n_h is the number of neurons in the hidden layer, z_i the output of each neuron of the hidden layer and γ is the activation function.

The advantages of MLPs are their capability to generalize and model nonlinear behaviours and the effective predictive performances, while the main cons are related to the complexity, since the hidden layers have a non-convex loss function which leads to different results depending on the weight initialization. MLPs also require the calibration of hidden neurons, layers and number of iterations and, even with small networks, it overparametrized models may be obtained. However, these overfitting phenomena can be mitigated by adding a penalty term to the loss function.

3.3 Radial Basis Function Network

A Radial Basis Function (RBF) Network is a particular type of artificial neural network that uses radial basis functions as activation functions. A radial basis function is a function that evaluates the distance between the input and a center point and returns a real value. A typical choice of distance metric is the Euclidean distance but other metrics can be chosen as well, depending on the application. The most commonly used radial basis function is the Gaussian:

$$r(|x-c|) = e^{-\frac{(x-c)^2}{2\rho^2}}$$

Notice that, in the above formulation, c is the center from which the Euclidean distance is evaluated and ρ determines the standard deviation of the function. Radial Basis Function Networks are widely used for different tasks such as prediction, classification and in particular function approximation, thanks to the very desirable properties of these functions such as continuity and differentiability. The output of the network is the result of the following linear combination:

$$\phi(x) = \sum_{i=1}^{N} a_i r(|x-c_i|)$$

where N is the number of radial basis functions and c_i and a_i are respectively the center and the weight of the $i-th$ function.

3.4 Gaussian Process Regression

Gaussian Process (GP) regression is a Bayesian machine learning approach, which is based on the hypothesis that the observations y are generated by the following model:

$$y(x) = f(x) + \varepsilon$$

where $x \in \mathbb{R}^m$ is the covariate vector, ε is a noise term with standard deviation σ_n and $f(x) \sim \mathcal{N}(m(x), k(x,x'))$

A Gaussian process is completely characterized by the mean and covariance function. Assuming, for the sake of simplicity, that $m(x) = 0$, the Gaussian process $f(x)$ is completely characterized by its covariance function, the so-called kernel, that has to be chosen properly in order to encode all prior knowledge.

There is a wide range of kernels that can be used depending on the situation (see [54], for an extensive review) and combinations can be also applied, e.g. through direct sum or tensor products.

Common choices are the squared exponential kernel

$$k_{se}(x,x') = \sigma_s^2 e^{\left(-\frac{\|x-x'\|^2}{2l^2}\right)}$$

and the cubic spline kernel

$$k_{sp}(x,x') = \sigma_f^2 \left(\frac{|x-x'|v^2}{2} + \frac{v^3}{3}\right)$$

where $v = min(x,x')$ while the signal variances σ_s^2, σ_f^2 and the length-scale l^2 are hyperparameters that need to be properly tuned, for example through marginal likelihood maximization.

Finally, given a new covariate vector x^*, the prediction $\hat{f}(x^*)$ can be computed as

$$\hat{f}(x^*) = \sum_{i=1}^{n} \alpha_i k(x(i), x^*)$$
$$\alpha = (K + \sigma_n^2 I)^{-1} y$$

3.5 Tikhonov regularization

Tikhonov regularization is a popular approach used to face ill-posed problems, by introducing some prior knowledge about the set of possible solutions. In linear regression, it is commonly used when the explanatory variables are linearly associated (multicollinearity), which can cause the estimate to have a high variance.

Let us consider the following linear model:

$$\mathbf{Y} = \mathbf{X}\theta + \epsilon \tag{3.1}$$

with $\mathbf{Y}$ observations vector, $\mathbf{X}$ matrix of covariates, θ parameter vector and ϵ noise term. In order to estimate θ, the traditional least squares optimization problem can be converted into a Tikhonov regularization-based estimation by adding a penalty to the cost function [55]:

$$\hat{\theta} = \arg\min_{\theta}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta) + \lambda\|\mathbf{\Gamma}\theta\|_2^2$$

where $\|.\|_2$ is the Euclidean norm and Γ is the Tikhonov matrix, whose choice determines which solutions are preferred, e.g. $\Gamma = I$ gives preference to solutions with smaller norms (this approach is called 'ridge' regression [56]). The hyperparameter λ, which needs to be properly tuned, determines how much regularization to enforce in the solution.

3.6 LASSO regularization

Consider again the model (3.1). The LASSO regularization is similar to the Tikhonov regularization but instead of using an Euclidean norm, it applies an l_1-norm:

$$\hat{\theta} = \arg\min_{\theta}(\mathbf{Y} - \mathbf{X}\theta)^T(\mathbf{Y} - \mathbf{X}\theta) + \lambda\|\theta\|_1$$

Despite being similar to the ridge regression, the usage of the l_1-norm in place of the l_2-norm allows the LASSO to set some parameters to zero.

For this reason the LASSO is typically used for variables selection in problem characterized by collinearity issues [57]. The main drawback of the LASSO is that its loss function is not differentiable, making the computation of its solution more complicated than the ridge one. However, a great variety of optimization methods have been developed to face this task.

3.7 Penalized smoothing splines

Splines are special functions able to model nonlinear behaviour while avoiding overfitting and boundary issues which are typical of high-order polynomials in regression problems. [58]. They are obtained by choosing a set of control points, called ‘knots’, that split the dataset into bins, and fitting a low-order polynomial within each interval, while applying continuity constraints at the knots.

Cubic splines in particular are piecewise cubic polynomials with continuous derivatives up to order 2 at each knot.

Let consider the observations y_i, and the corresponding predictors x_i, $i = 1, ..., N$. The regression cubic spline $\hat{f}$ is the solution of the following optimization problem:

$$\hat{f} = \arg\min_{\tilde{f}} \sum_{i=1}^{N} \left(y_i - \tilde{f}(x_i)\right)^2$$

$$\tilde{f} = \sum_{j=1}^{K} \beta_j h_j(x) \tag{3.2}$$

where $\tilde{f}$ is as in (3.2), β_j are the spline coefficients, K is the number of knots and $h_j(x)$ are the basis functions. Although there is no unique choice for the basis functions, the B-splines are widely used in the literature because of their computational properties [59].

There is a variety of strategies for enforcing smoothness to the fitted curve, for example reducing the number of knots. A typical choice is to keep a high number of knots and add a penalty term to the previous optimization problem, which becomes

$$\hat{f} = \arg\min_{\tilde{f}} \sum_{i=1}^{N} \left(y_i - \tilde{f}(x_i)\right)^2 + \lambda \int_{-\infty}^{\infty} \left(\tilde{f}''(x)\right)^2 dx$$

where λ is the regularization parameter.

The result of the above optimization is the cubic penalized B-spline (cubic P-spline) [60]. An additional constraint can be applied to the spline basis in order to enforce continuity between the first and the last boundary of the considered independent variable. This leads to the cyclic cubic P-splines, which is useful whenever it is necessary to model periodic behaviours within the considered variables (e.g. day of year or time of day) [61].

4

Spectral characterization of the multi-seasonal component of the Italian electric load: a LASSO-FFT approach

The present chapter is focused on a fundamental ingredient of long-term predictors, that is the the multi-seasonal component of the load time series. While the daily frequency is a multiple of the weekly one, both the daily and weekly frequencies are not exact multiple of the yearly one, which is equal to the inverse of 365.25 days. As a consequence of the interplay between weekly, yearly and leap-year periodicity, the multi-seasonal component is periodic with period 28 years, since the calendar repeats with the same weekday-date combination every 28 years. Concerning the multi-seasonal component, different solutions have been proposed in the literature both for short-term forecasts (with model strategies such as dummy variables [62], neural networks [63], Seasonal Exponential Smoothing [64], [65], [66]) and long-term ones [67], [68], typically assuming an additive relation between the different periodicities of the signal. However, the additive model does not allow for modifications of the daily profile depending on the different seasons of the year, implying a lack of flexibility at both country [69], [70] and individual customers level [71]. The latter contribution adopted a multi-task machine learning approach, while the former one relied on a Fourier series parametrization, where the relevant harmonics and the possible interaction terms between pairs of harmonics were selected via stepwise regression.

Here, in the context of a full parametrization of the multi-seasonal component in the Fourier domain, that for quarter-hourly data involves almost 1 million parameters, we address two main issues. The first one is the rigorous detection of interaction terms between yearly and weekly harmonics. The second issue is the development of a numerically efficient procedure to obtain a sparse parametrization in the Fourier domain. At first sight, Fast Fourier Transform methods are not applicable because the complete 28-year series is usually not available and, moreover, holidays (during which the power demand may substantially depart from the multi-seasonal pattern) must be treated as missing data. Compared to existing works in the field of spectral analysis of nonuniformly sampled data (see [72], [73]), a major contribution of the paper is the development of a novel Least Absolute Shrinkage and Selection Operator with Fast Fourier Transform (LASSO-FFT) estimator that yields a sparse solution in $O\left(n\log(n)\right)$ operations.

The new identification algorithm has been validated on the quarter-hourly Italian electric load demand collected from 1990 to 2015. The sparse parametrization provided by the LASSO estimator reduces the number of parameters from 981,792 to 711. The periodic structure of the Italian demand explains a substantial fraction of the load profile, a somehow unexpected feature that may prove useful in several respects. In particular, since the multi-seasonal pattern remains remarkably stable throughout the years, the new method paves the way for the design of mid-, long- and very long-term forecasters.

4.1 Dataset and problem statement

The time series of the Italian electric load coming from a wide historical database covers a period of 26 years, from 1990 to 2015, each sample representing the average power consumption during a 15 min interval, see Fig. 4.1, top panel. The time series is clearly non-stationary both in mean and variance. A preliminary logarithmic transformation [25], [74], followed by detrending has been employed to transform the data into a multi-seasonal series. More precisely, the following model for the load $L(t)$ is assumed:

$$L(t) = \exp\left(T(t) + F(t) + \sum_i H_i(t) + e(t)\right)$$

where $T(t)$ denotes the trend, $F(t)$ is the deterministic multi-seasonal component including daily, weekly and yearly periodicities, $H_i(t)$ are the so-called intervention events accounting for the effect of holidays, and $e(t)$ is an error term, often modeled as a stationary random process.

Following [69], the trend $T(t)$ was estimated as the output of a lowpass filter with bandwidth 1/730 day^{-1}. As it can be seen in the center and bottom panel of Fig. 4.1, the log-transformed and detrended series

$$y(t) = \ln\left(L(t)\right) - T(t) = F(t) + \sum_i H_i(t) + e(t)$$

exhibits a relatively stable profile, whose variability is mostly due to the simultaneous presence of daily, weekly and yearly seasonalities. If the 26 yearly profiles, properly shifted to align weekdays, are overlapped, one can visually assess the remarkable regularity of the multi-seasonal components, with the exception of special days, i.e. holidays (in red), whose loads are far less predictable, see Fig. 4.1, middle panel. In fact, most holidays are associated with fixed dates (e.g. December 25) so that the weekday changes from year to year. In the deterministic multi-seasonal component $F(t)$, three periodic components are present: yearly, weekly and daily. In view of leap years the actual yearly period is 365.25 days. As recalled in the Introduction, the multi-seasonal component $F(t)$ is periodic with period 28 years. Under a quarter-hourly sampling, this amounts to $28 \times 365.25 \times 96 = 981,792$ unknowns.

A suitable parametrization for the multi-seasonal component is a Fourier expansion,

$$F(t) = \sum_{i=1}^{2N_w(1+2N_y)} \theta_i h_i(t), \quad h_i(t) \in \mathcal{Y} \otimes \mathcal{W}$$

$$\mathcal{Y} = \{\cos(j\Psi t), j \in [0, N_y]\} \cup \{\sin(j\Psi t), j \in [1, N_y]\}$$

$$\mathcal{W} = \{\cos(k\Omega t), k \in [0, N_w]\} \cup \{\sin(k\Omega t), k \in [1, N_w - 1]\}$$

$$\Psi = \frac{2\pi}{365.25}, \quad \Omega = \frac{2\pi}{7}$$

Therefore, h_i is the product of a 7-day and a 365.25-day harmonic, $N_y = 730$ and $N_w = 336$ being the number of yearly and weekly harmonics, respectively. The total number of Fourier parameters, $2N_w(1 + 2N_y) = 981,792$, coincides with the parametrization in the time-domain. The multi-seasonal component is expected to exhibit a smooth profile, the type of signal that can be adequately captured by a sparse parametrization in the Fourier domain.

Multi-seasonal component estimation problem: given $y(t)$, $t_0 \leq t \leq t_f$, identify a sparse model in the Fourier domain for the multi-seasonal component $F(t)$.

4.2 Estimating the multi-seasonal component

4.2.1 Additive vs interaction model

A rather obvious way to reduce the complexity of the multi-seasonal component model is to resort to an additive model [67], [68]:

$$F(t) = F_y(t) + F_w(t)$$

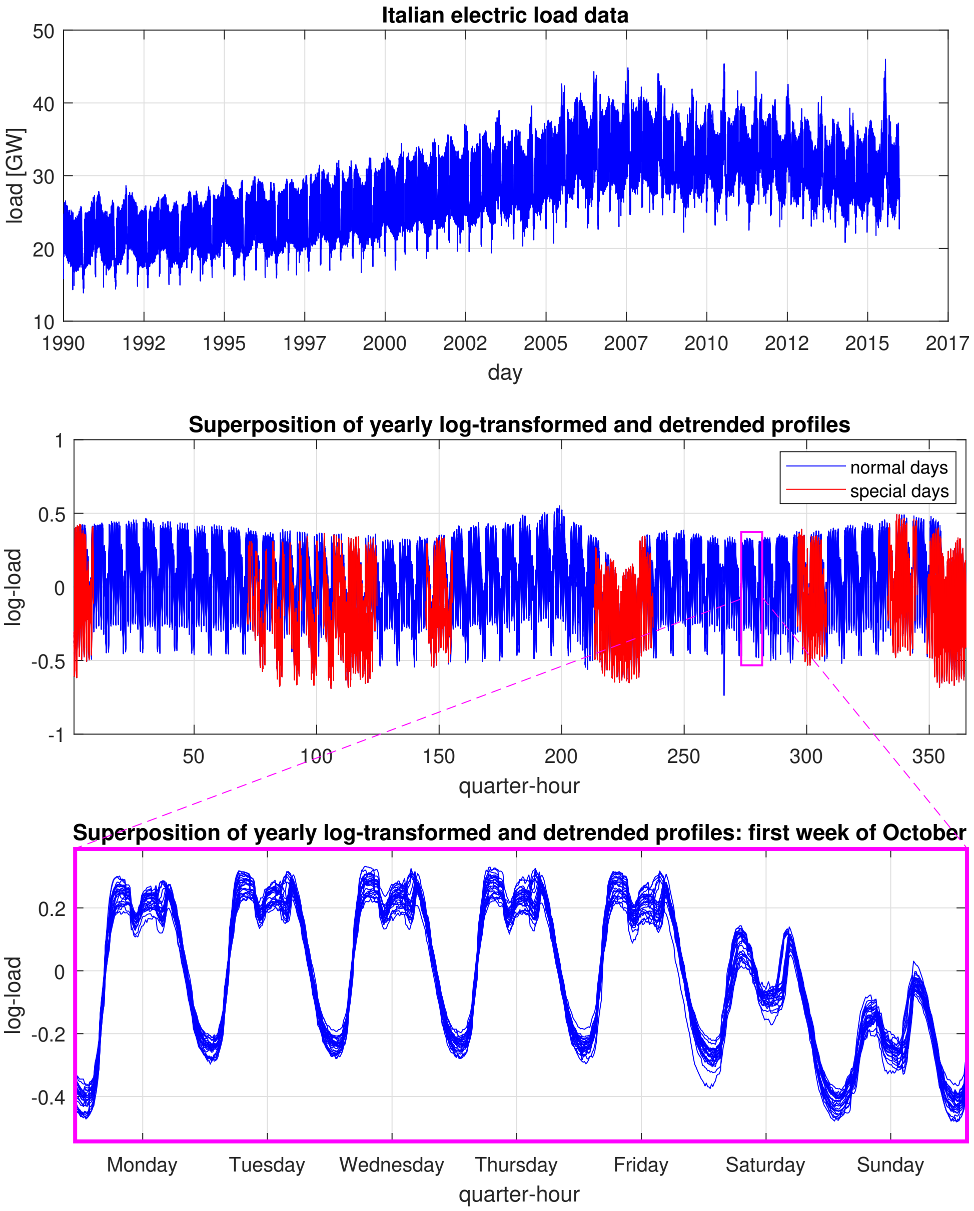


FIGURE 4.1: Time series of Italian daily power consumption in the period 1990-2015 (top panel). Superposition of 26 yearly log-transformed and detrended profiles, realigned to match weekdays (middle and bottom panel).

where F_y is periodic of period 365.25 and F_w is periodic of period 7 (note that this includes also the daily periodicity). In this additive model, the restrictive assumption is made that the weekly (and daily) demand patterns remain unchanged throughout the year. In other terms, it does not account for the presence of interactions between the yearly and weekly harmonics [69], [70]. The existence of such interactions can be visually ascertained by exploiting a well known trigonometric equivalence. Indeed, the products of different periodicities leads to 'fringe' harmonic components around the main ones, as a consequence of trigonometric Werner formulas:

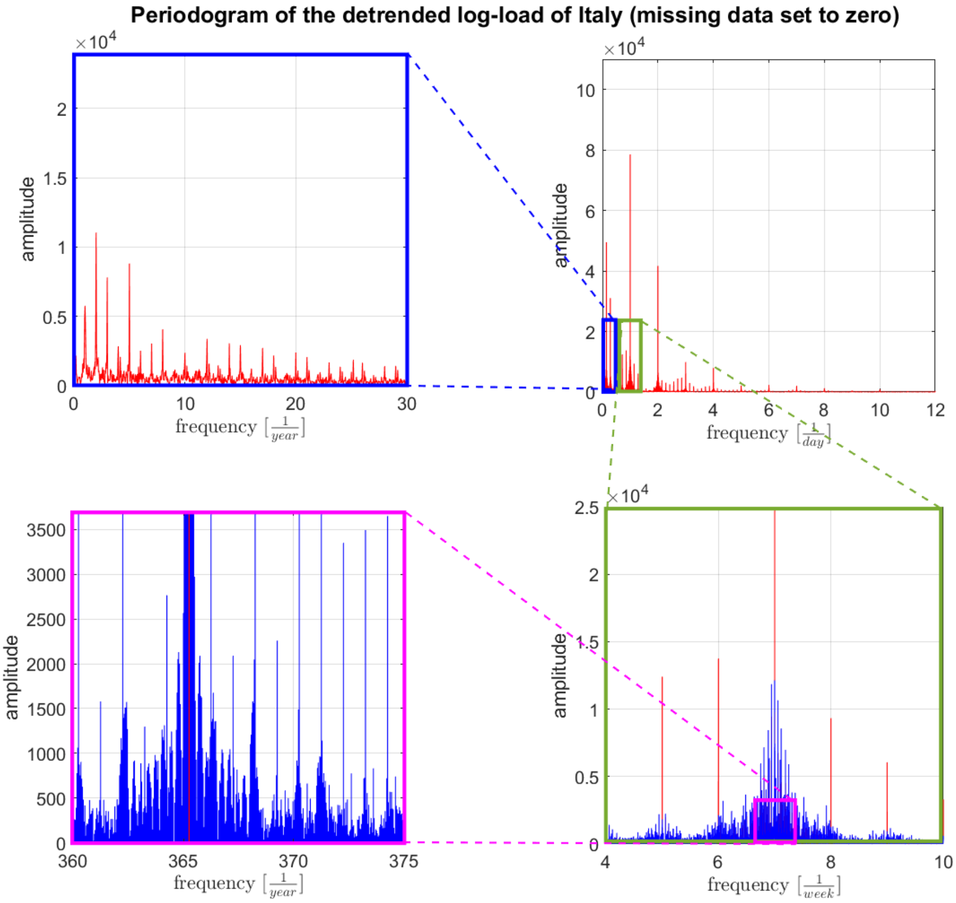


FIGURE 4.2: Periodogram of the log-transformed and detrended load. Load values in correspondence of intervention events have been set to zero and this introduces some distortion in the estimated spectrum. Nevertheless, the presence of several relevant 'fringe' harmonic components due to interactions between yearly, weekly and daily periodicities is noticeable.

$$\sin(\alpha)\cos(\beta) = \frac{1}{2}\left[\sin(\alpha+\beta) + \sin(\alpha-\beta)\right]$$

$$\cos(\alpha)\cos(\beta) = \frac{1}{2}\left[\cos(\alpha+\beta) + \cos(\alpha-\beta)\right]$$

$$\sin(\alpha)\sin(\beta) = \frac{1}{2}\left[\cos(\alpha-\beta) - \cos(\alpha+\beta)\right]$$

In Fig. 4.2, the periodogram of $y(t)$ is displayed. There is a clear evidence of the existence of fringe harmonics around the weekly ones. It is also seen that the distance between two adjacent fringe harmonics is just 1/year, as predicted by the Werner formulas. In view of this, the interaction terms should not be dropped in the model and other ways to achieve sparsity should be looked for.

4.2.2 Ill posedness of the estimation problem

If the load had been observed over a whole 28-year interval, a standard FFT (Fast Fourier Transform) approach could be employed in order to efficiently compute the component. Let $\theta \in \mathbb{R}^n$, $n = 981,792$, denote the vector containing the coefficients of the Fourier expansion of $F(t)$. Then,

$$Y = \Phi\theta + \epsilon$$

where $Y = \begin{bmatrix} y(1) & y(2) & \dots & y(n) \end{bmatrix}^T$ is the vector of log-transformed and detrended loads, and the columns ϕ_i of $\Phi \in \mathbb{R}^{n\times n}$ are the sampled regressors, i.e.

$$\phi_i = \begin{bmatrix} h_i(1) & h_i(2) & \dots & h_i(n) \end{bmatrix}^T$$

Finally, ϵ is a noise term. Consider the least squares estimate θ^{LS}. In view of the orthogonality properties of the sinusoidal and cosinusoidal regressors:

$$\theta^{LS} = \left(\Phi^T\Phi\right)^{-1}\Phi^T Y = \Phi^T Y = \Phi^{-1} Y$$

Observe that $\Phi^T Y$ is just the Discrete Fourier Transform (DFT) of Y which can be computed with $O\left(n\log(n)\right)$ complexity via the FFT algorithm. Moreover, in view of the orthogonality of the regressors, model order reduction can be performed without need of re-estimating the model parameters. For instance, if a subset of harmonics can be selected by stepwise regression, their Fourier coefficients are those estimated for the full model, irrespective of the number of harmonics included in the considered model.

There are two reasons, however, that prevent a direct FFT approach. The first one is the unavailability of the whole 28 year dataset. For instance, the Italian load data, ranging from 1990 to 2015, cover just 26 years. The second one has to do with holidays, in correspondence of which a substantial departure from the multi-seasonal pattern is usually observed. As far as the estimation of the multi-seasonal component is concerned, these special days can be regarded as missing data and discarded from the observation vector. Therefore, the available dataset is given by $\tilde{Y} = SY$, where $\tilde{Y} \in \mathbb{R}^{\tilde{n}}$ with $\tilde{n} < n$ and $S = \begin{bmatrix} S_1 & S_2 & \dots & S_{\tilde{n}} \end{bmatrix}^T \in R^{\tilde{n}\times n}$ is a selection matrix with:

$$S_i = \mathbf{e_j}, \quad \text{where } j \text{ is the } i\text{th non-missing data}$$

In matrix form,

$$\tilde{Y} = S\Phi\theta + \epsilon = \tilde{\Phi}\theta + \epsilon, \quad \tilde{\Phi} = S\Phi \tag{4.1}$$

Not only the new problem is underdetermined (there are more unknowns than data), but the columns of $\tilde{\Phi}$ are no more guaranteed to be orthogonal. This means that, even if ill posedness is addressed by means of regularization techniques, such as the LASSO discussed below, the computational burden of finding a solution in a very large dimensional space is not straightforwardly alleviated via FFT-based algorithms. This motivates the interest for the two problems addressed in the rest of the chapter:

Problem 1: Identify a sparse solution of the ill posed parameter estimation problem (4.1).

Problem 2: Devise a computational algorithm capable of obtaining the sparse solution efficiently.

4.3 Sparse regularization

In order to overcome ill posedness one can introduce a regularization term in the Least Squares problem formulation. In particular, in order to enforce sparsity, the use of the LASSO is explored [57]. Given a general linear model:

$$Y = X\theta + \epsilon$$

with $Y \in \mathbb{R}^{n\times 1}$, $X \in R^{n\times m}$, $\theta \in \mathbb{R}^{m\times 1}$ and ϵ a noise term, the LASSO regularized estimate is defined as:

$$\theta^{\text{LASSO}} = \arg\min_{\theta} \left(\frac{1}{n} \|X\theta - Y\|^2 + \lambda \|\theta\|_1 \right)$$

The balance between the loss function and the regularization term is controlled by the scalar regularization parameter λ.

Among the great variety of methods adopted to optimize the LASSO objective function, such as least-angle regression (LARS), homotopy method, Alternating Direction Method of Multipliers (ADMM) [75], [76], [77], one of the most popular technique is the proximal gradient descent or Iterative Shrinkage-Thresholding Algorithm (ISTA) [78], [79]. In large scale problems, gradient-based algorithms are preferred to more sophisticated approaches because of their simplicity, as pointed out in [79]. However, ISTA methods are also known to be slow [80]. Below, we develop a new method for the LASSO estimation of multiperiodic signals, which exploits the FFT to compute the gradient of the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [79] [81].

The k-th iteration step of the accelerated gradient descent method is given by

$$\theta_k = \text{Prox}_{\gamma\lambda}\left(v_k - \gamma \nabla E\left(v_k\right)\right) \tag{4.2}$$

$$\nabla E(v_k) = \frac{2}{n}(X^T X v_k - X^T Y) \tag{4.3}$$

$$v_k = \theta_{k-1} + \frac{\tau_{k-1} - 1}{\tau_k}(\theta_{k-1} - \theta_{k-2}) \tag{4.4}$$

$$\tau_k = \frac{1 + \sqrt{1 + 4\tau_{k-1}^2}}{2} \tag{4.5}$$

$$\text{Prox}_{\gamma\lambda}(\theta_k) = \begin{cases} \theta_k - \gamma\lambda, & \text{if } \theta_k > \gamma\lambda \\ 0, & \text{if } \theta_k \in [-\gamma\lambda, \gamma\lambda] \\ \theta_k + \gamma\lambda, & \text{if } \theta_k < -\gamma\lambda \end{cases} \tag{4.6}$$

Above, $E(\theta) = \frac{1}{n}\|X\theta - Y\|^2$ is the loss function, $\nabla E(v_k)$ is the descent direction, v_k and τ_k are the momentum term and its coefficient [81], γ is the step size, and $\text{Prox}_{\gamma\lambda}$ is the so-called proximal operator. The initialization θ_0 is randomly chosen and τ_0 is set equal to 1 so that $v_1 = \theta_0$. The algorithm terminates when $\|\theta_k - \theta_{k-1}\| < tol$, where tol is a sufficiently small positive constant (herein: $tol = 10^{-6}$). Once convergence has been reached, the non-zero elements in θ^{LASSO} are re-estimated via Least Squares.

When applied to model (4.1), the expression (4.2) yields

$$\theta_k = \text{Prox}_{\gamma\lambda}\left(v_k - \frac{2\gamma}{\tilde{n}}\left(A\left(v_k\right) - B\right)\right)$$

$$A(v_k) := \Phi^T S^T S \Phi v_k, \qquad B := \Phi^T S^T \tilde{Y}$$

Note that $\bar{Y} := S^T\tilde{Y}$ is just the 28-year series that is obtained from $\tilde{Y}$ when missing data are filled with zeros. Since B is just the Discrete Fourier Transform (DFT) of $\bar{Y}$, it can be efficiently computed via FFT. Similarly, $A(v_k)$ is the DFT of $\hat{Y}_k = S^T S \Phi v_k$. Observing that Φ is just an Inverse Discrete Fourier Transform (IDFT) operator in matrix form, also $\hat{Y}_k$ can be efficiently computed. First, the IDFT of v_k is calculated and then all the entries of the resulting 28-year series corresponding to missing data are zeroed. In conclusion, exploiting FFT algorithms, each FISTA iteration can be efficiently executed with $O\left(n \log(n)\right)$ complexity.

Algorithm LASSO-FFT algorithm

```
randomly initialize $\theta_0$ and set $\tau_0 = 1$ and $\theta_{-1} = 0$
$\bar{Y} \leftarrow S^T \tilde{Y}$
$B \leftarrow \text{FFT}(\bar{Y})$
$k \leftarrow 0$
while $\|\theta_k - \theta_{k-1}\| > tol$ do
    $\theta_{k-2} \leftarrow \theta_{k-1}$
    $\theta_{k-1} \leftarrow \theta_k$
    $\tau_k \leftarrow \frac{1+\sqrt{1+4\tau_{k-1}^2}}{2}$
    $v_k \leftarrow \theta_{k-1} + \frac{\tau_{k-1}-1}{\tau_k}(\theta_{k-1} - \theta_{k-2})$
    $\hat{Y}_k \leftarrow S^T S\ (\text{IFFT}(v_k))$
    $A_k \leftarrow \text{FFT}(\hat{Y}_k)$
    $\theta_k \leftarrow (v_k - \gamma\,(A_k + B))$
    $\theta_k\,(\theta_k > \gamma\lambda) \leftarrow \theta_k - \gamma\lambda$
    $\theta_k\,(\theta_k < -\gamma\lambda) \leftarrow \theta_k + \gamma\lambda$
    $\theta_k\,(\theta_k \in [-\gamma\lambda, \gamma\lambda]) \leftarrow 0$
    $k \leftarrow k + 1$
end while
```

4.4 Calibration

4.4.1 Gradient step size

Recall that, if $E(\theta)$ is convex, differentiable and $\nabla E(\theta)$ is Lipschitz continuous with Lipschitz constant L, the momentum-based gradient descent with fixed stepsize $\gamma \leq 1/L$ is guaranteed to converge [79]. In view of this, a condition for the convergence of the LASSO-FFT can be given.

Theorem. if $\gamma \leq \tilde{n}/2$ the LASSO-FFT is guaranteed to converge.

Proof. Observe that

$$\nabla E(\theta_2) - \nabla E(\theta_1) = \frac{2}{\tilde{n}} \Phi^T S^T S \Phi\,(\theta_2 - \theta_1)$$

$$\frac{\|\nabla E(\theta_2) - \nabla E(\theta_1)\|}{\|\theta_2 - \theta_1\|} \leq \frac{2}{\tilde{n}} \|\Phi^T S^T S \Phi\|$$

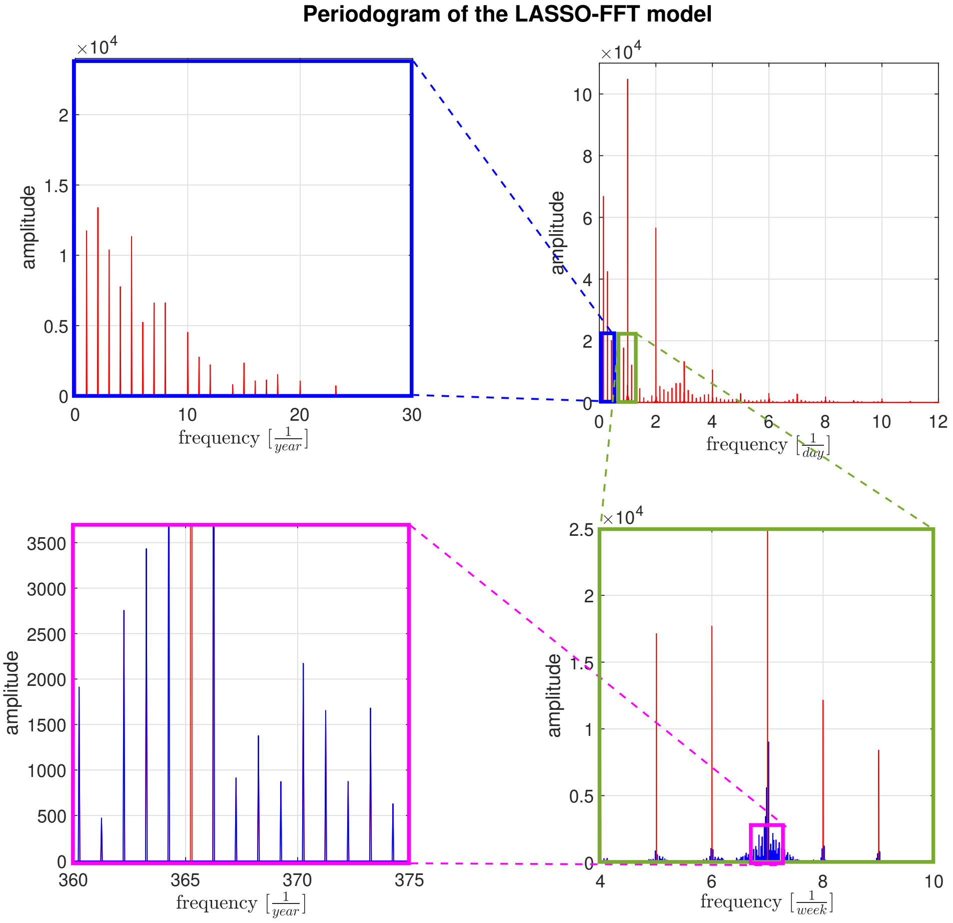


FIGURE 4.3: Periodogram of the sparse model obtained through the LASSO-FFT algorithm. In the third and fourth panel the blue harmonics are due to the interaction of daily and yearly periodicity.

Since $\|S^T S\| \leq \|I_n\|$ (where the equality holds for $\tilde{n} = n$),

$$\|\Phi^T S^T S \Phi\| \leq \|\Phi^T I_n \Phi\| = \|I_n\| = 1$$

Hence, $2/\tilde{n} \geq L$, Q.E.D.

4.4.2 Regularization parameter

Let $\mathcal{T}$ denote the set of time indices associated with the data vector $\tilde{Y}$ used to estimate the multi-seasonal component. The metric used to assess performance is the restricted quarter-hourly Mean Absolute Percentage Error (MAPE):

$$\overline{MAPE} = \frac{100}{\tilde{n}} \sum_{t \in \mathcal{T}} \left| \frac{\hat{L}(t) - L(t)}{L(t)} \right|$$

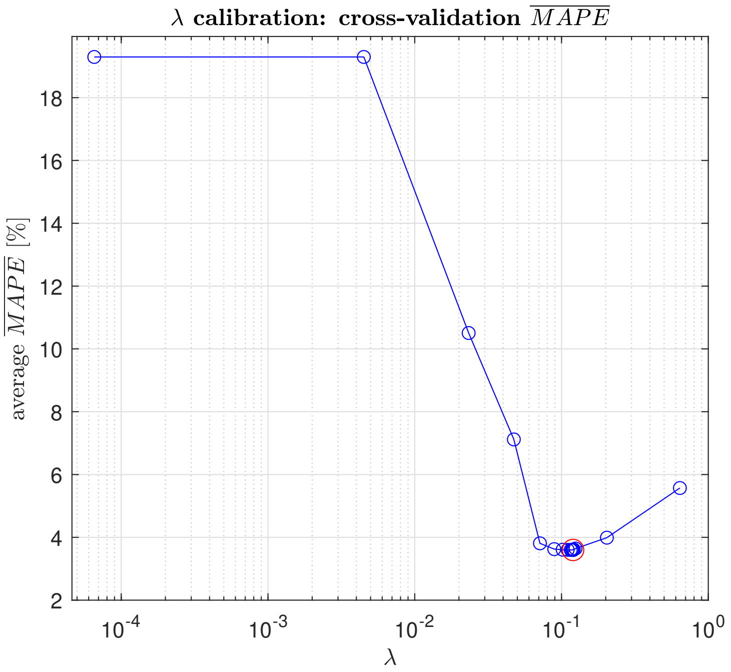


FIGURE 4.4: Cross-validated average $\overline{MAPE}$ as a function of the LASSO regularization parameter λ. The minimum (thick circle) was found by golden-section search.

$$\hat{L}(t) = \exp\left(\hat{T}(t) + \hat{F}(t)\right)$$

where $\hat{F}(t)$ denotes the estimated multi-seasonal function and $\hat{T}(t)$ the predicted trend component, which is taken constant and equal to the last available trend value of the training set. In fact, for a one-year ahead prediction the trend can be regarded as reasonably close to the terminal value of the previous year's trend. For longer horizons, the prediction of the trend cannot be based on past load data alone, but calls for a forecast of future economic scenarios, see [17].

The regularization parameter was tuned through rolling forecast origin cross-validation [82]. A prediction horizon of one year was considered and each time the model was trained on all the data of the years preceding the validation one, with a minimum of 5 years (1990-1994) of data for training the model that generated the first prediction. This implied that 21 instances of restricted MAPE were averaged for each candidate regularization parameter. The experiment yielded to $\lambda = 0.12$ as value with minimum average restricted MAPE, Fig. 4.4.

4.5 Computational effort

In order to assess how the computational efficiency scales with the dimension of the problem, the algorithm was applied to the Italian load data with different sampling rates, ranging from quarter-hourly to 1/21 days. The results, displayed in Fig. 4.5, show that, with a very good approximation, the computational time scales as $O(nlog(n))$ (continuous regression line through

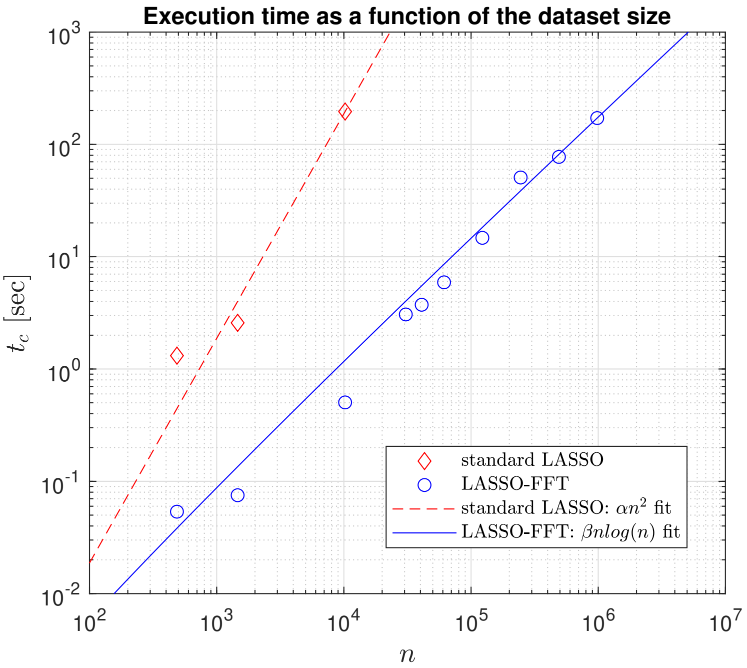


FIGURE 4.5: Execution time vs dataset size n: comparison between the LASSO-FFT (circles) and the standard implementation of the LASSO (diamonds). Continuous line: least squares fitting of $\beta n \log(n)$; dashed line: least squares fitting of αn^2.

circles) and is much lower than the computational time required by the standard implementation of the LASSO algorithm (dashed regression line through diamonds, scaling as $O(n^2)$). The new method is more than 10 times faster for $n \sim 1,000$ and more than 100 times faster for $n \sim 10,000$. In particular, for the size of the considered problem ($n = 10^6$), the proposed approach yield to an execution time of ~ 2 minutes, while the standard LASSO execution time can be predicted to be between 11 and 12 days.

4.6 Use for long-term forecasting

When applied to the Italian load data, the LASSO yielded a 711-dimensional model, corresponding to a reduction ratio of $711/981,792 = 7.2 \times 10^{-4}$. The spectrum of the estimated multi-seasonal component, shown in Fig. 4.3, reveals two remarkable features of the multi-periodic component of the Italian load demand. First, the yearly periodicity pattern is well captured by the first 20 yearly harmonics, see Fig. 4.3, blue box. Second, the, fringe interaction harmonics are present mainly around the daily harmonics rather than around the weekly ones, see Fig. 4.3, green and purple boxes. This means that, compared to the the weekly pattern of daily loads, it is the intra-day pattern of quarter-hourly loads that is more subject to seasonal changes during the year. In order to assess the effectiveness of the sparse model as a long-term predictor, the performances of one-year ahead forecasts were evaluated over the years 2013-2014-2015 and compared with the traditional additive model proposed in other works [68]. For each predicted year, all load data until the test year were used as training set. The MAPEs on the test data (Table 4.1) prove the superiority of the multi-seasonal multiplicative model.

The prediction performances during the first weeks of October 2014 and October 2015 can be visually appreciated in Fig. 4.6. Apparently, a large fraction of load variability is captured by the proposed multi-seasonal model, as confirmed by the R^2 coefficients, both above 0.9.

In order to better highlight the performances of the presented long-term forecaster, a more challenging touchstone has been used, i.e. the performances of short-term predictors employed by the Italian TSO. In particular, the one-day ahead predictions of daily loads provided by the Sibilla and Trinity predictors were compared with the daily load forecasts obtained by summing up the one-year ahead quarter-hourly loads predicted by the LASSO model, see Table 4.2. It is remarkable that, in spite of the different prediction horizon, the one-year ahead restricted MAPE is only 1.72 times larger than the best one-day ahead MAPE.

4.7 Discussion

In this chapter an efficient $O\left(n\log(n)\right)$ LASSO algorithm for the identification of a sparse model of the multi-periodic component of the electrical load has been proposed and tested on 1990-2015 Italian data. As an alternative one could resort to a stepwise scheme (like in [69]), which, besides being computationally demanding, would guarantee only a suboptimal solution. The results demonstrate that the multi-seasonal component is remarkably stable through the years and accounts for a significant fraction of the load. The application of the estimated multi-seasonal component for long-term forecasting has been illustrated, while future developments could regard its use also within short-term predictors.

Table 4.1: Restricted MAPE: comparison between the LASSO-FFT model and the LASSO-FFT additive model.

	2013	**2014**	**2015**
LASSO-FFT	4.02%	2.92%	4.28%
LASSO-FFT (additive)	5.95%	5.46%	6.57%

Table 4.2: Comparison between two predictors used by the TSO and the LASSO-based model for the 2013 one-year ahead forecast.

	Pred. horizon	**MAPE$_{\textbf{DAILY}}$**
Sibilla (all days)	one day	2.30%
Trinity (all days)	one day	2.10%
LASSO-FFT (normal days)	1 year	3.63%

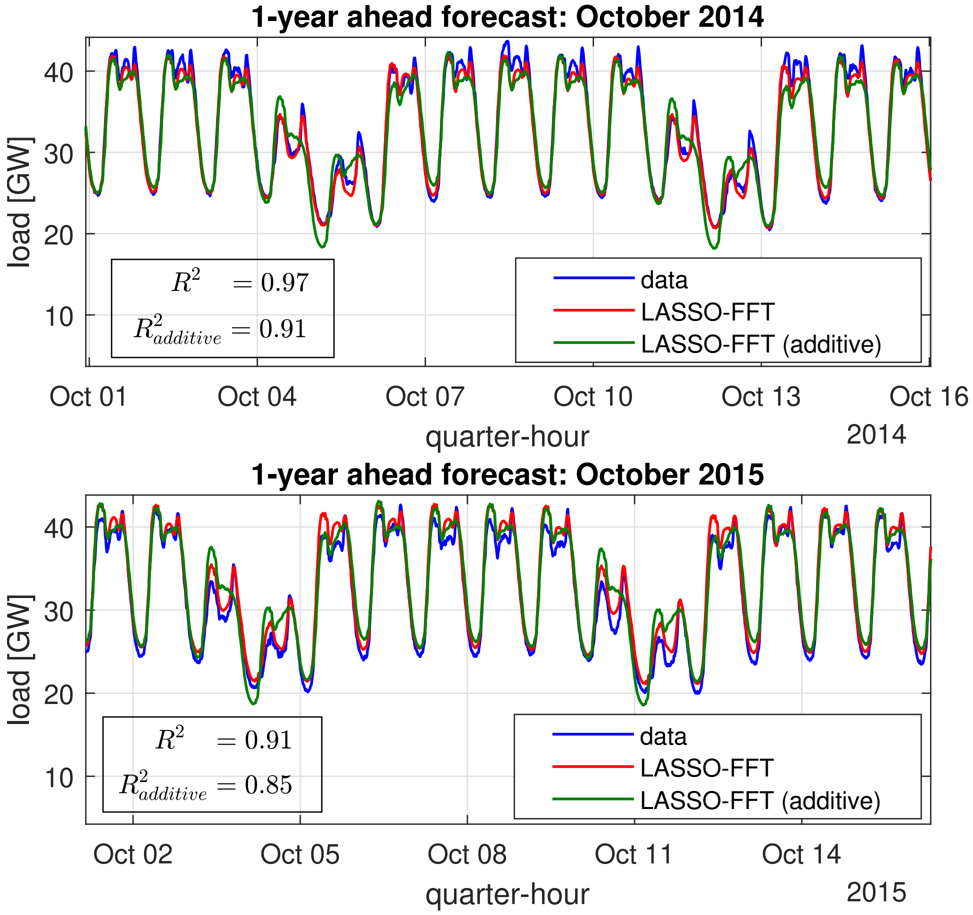


Figure 4.6: First half of October: one-year ahead forecasts for 2014 (top) and 2015 (bottom).

5

Regularization methods for the short-term forecasting of the Italian electric load

The purpose of this chapter is the development of a whole-day ahead forecast for the quarter-hourly electric load of Italy during normal days by relying exclusively on the loads recorded in the previous days. This can be done for the load demand time series, without resorting to exogenous data such as weather forecasts, in view of its highly correlated nature [83], [84], [85], [86]. The analysis is restricted to so-called 'normal days', that is those days without special events such as holidays. It will be shown that, after a suitable data preprocessing, consisting of a logarithmic transformation and a 7-day differentiation, accurate one-day ahead predictions of the 24-hour profile can be achieved via a weighted linear combination of the 24-hour profile of the previous day. In view of the quarter-hourly sampling, the 24-hour profile is represented by a vector of 96 loads. Therefore, the identification of the predictor weights involves the estimation of a 96×96 parameter matrix. In view of the strong correlation between consecutive quarter-hourly load values, it is reasonable to assume that the entries of the matrix behave as a smooth surface, an observation that justifies the adoption of regularization-based machine learning techniques. Different strategies are applied in order to reduce the degrees of freedom and the variance of the parameters. The obtained forecasts are compared over different test years, using as state-of-art benchmark the predictor of the national Transmission System Operator (TSO) Terna.

The results show that the proposed solutions achieve significant improvements, in terms of Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) and Mean Squared Error (MAE) with respect to the Terna forecaster.

The chapter is organized as follows: in Section 5.1 the dataset and the preprocessing phase are described, while in Section 5.2 the problem statement is formulated. Section 5.3 presents

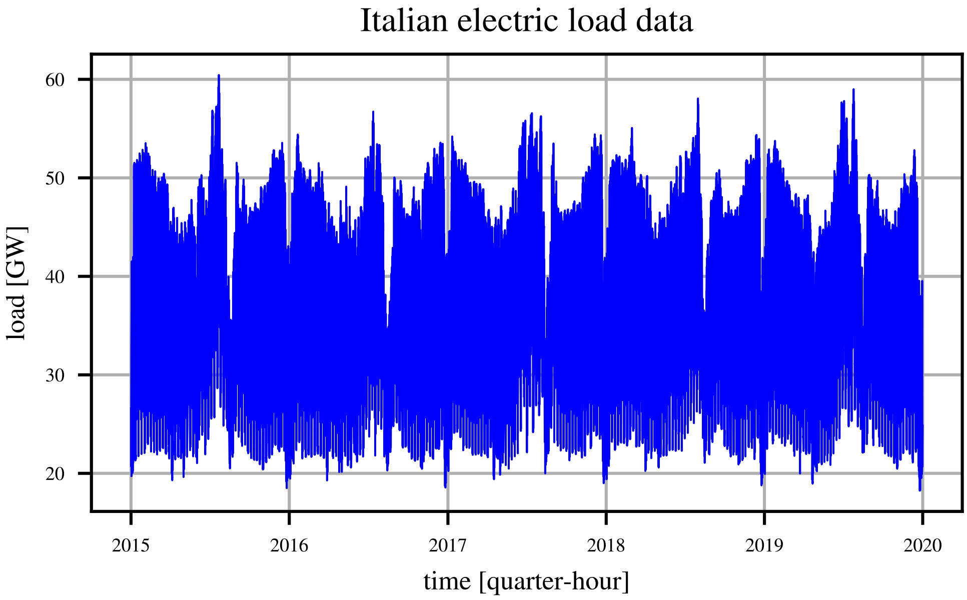


FIGURE 5.1: Italian quarter-hourly electric load demand from 2015 to 2019.

the different modelling techniques adopted and Section 5.4 describes the experiment setups. Section 5.5 compares and discusses the predictive performances of the proposed techniques. Finally, Section 5.6 summarizes the main results and concludes the chapter.

5.1 Dataset and preprocessing

The available data consist of: (i) a 5-year long time series of quarter-hourly Italian electric load demands (from 2015 to 2019); (ii) a 3-year long time series of quarter-hourly forecasts elaborated by the national Transmission System Operator (TSO) Terna (from 2017 to 2019) [1]. Both datasets were downloaded from the Transparency Report Platform of Terna available at [87].

The Italian electric load demand is displayed in Fig. 5.1, while Fig. 5.2 displays an example of one-day ahead predictions by Terna over a week. It can be seen that, although the performance is rather good, there might still be some room for improvement, which motivates the analysis of this chapter.

In the following, $L(d, q)$ denotes the country load demand at the q-th quarter-hour of the day d, where $1 \leq q \leq 96$ and d is an integer serial representing the whole number of days from a fixed, preset date (e.g. January 0, 0000) in the proleptic ISO calendar.

[1] The 2015 and 2016 forecasts were also available on the Terna Transparency Report Platform. However, their forecasting error appears significantly biased, making them unusable for benchmarking purposes.

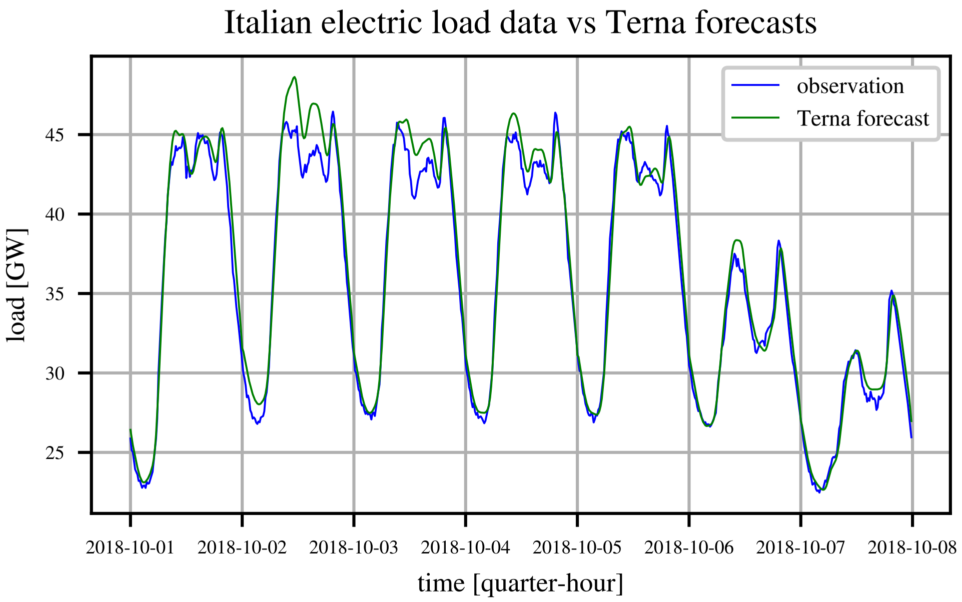


FIGURE 5.2: Italian quarter-hourly electric load demand vs Terna prediction over the first week of October 2018. In spite of the good accuracy, there is room for improvement as seen on Tuesday and Wednesday, where the actual load value is underestimated.

In the following, when referring to the signal $L(d,q)$, we will mean the univariate time series

$$\Big\{ \ \dots \quad L(1,1) \quad L(1,2) \quad \dots \quad L(1,96) \quad L(2,1) \quad \dots \quad L(2,96) \quad \dots \ \Big\}$$

Before proceeding with the implementation of the predictive model, a suitable preprocessing step is performed in order to obtain a signal that can be forecast more effectively. A rather common step is resorting to a logarithmic transformation of the data [74], [88],

$$S(d,q) := \ln\left(L(d,q)\right)$$

which results in the time series displayed in Fig. 5.3 top. For a short-term prediction purpose, low frequency components such as trend and yearly periodicities can be neglected, while faster phenomena such as weekly seasonalities remain relevant and must be taken into account. In particular, weekly periodicity is modelled by assuming that, on a short range framework,

$$S(d,q) = p(d,q) + \eta(d,q)$$

where $p(d,q) = p(d+7,q), \forall q$, is a deterministic periodic function in the first argument with period $T = 7$ days and the time series $\eta(d,q)$ is a zero-mean stationary stochastic process, with the exception of the so called ‘intervention events’, i.e. special days such as holidays, during which the typical weekly pattern is altered. In order to filter the weekly periodicity of the signal, a 7-day differentiation is applied to the log-transformed time series:

$$\tilde{Y}(d,q) := S(d,q) - S(d-7,q) = \eta(d,q) - \eta(d-7,q)$$

The resulting time series $\tilde{Y}(d,q)$, shown in Fig. 5.3 bottom, can be considered, in a short timespan, to be zero-mean and stationary, with the exception of special days that are excluded from this analysis, since they need an *ad hoc* modelling strategy.

In this preprocessing phase, the logarithmic transformation applied before the 7-day difference operator is crucial in order to make the marginal distributions of the data on each quarter-hour less skewed and closer to Gaussianity, which would allow an effective adoption of simple linear predictive models to achieve high forecasting performances.

Let us call $\mathcal{D}_s$ the set of all the special days (see Appendix A) and let define the series of 'cleaned' 7-day difference of log-load values as:

$$Y(d,q) = \begin{cases} \text{missing}, & \text{if } (d \in \mathcal{D}_s) \text{ or } (d-7 \in \mathcal{D}_s) \\ \tilde{Y}(d,q), & \text{otherwise} \end{cases}$$

5.2 Problem statement

The main objective of this chapter is the development of a one-day ahead forecaster $\hat{L}(d,q)$ for the daily profile of the Italian electric load $L(d,q), 1 \leq q \leq 96$, based on the knowledge of $L(t,q), \forall t < d, \forall q$. For the subsequent analysis, it is convenient to introduce the following lifted representation of the signals:

$$\mathbf{L}(d) = \begin{bmatrix} L(d,1) & L(d,2) & \dots & L(d,96) \end{bmatrix}^T$$

According to the lifted notation,

$$\mathbf{L}(d) = e^{\left(\tilde{\mathbf{Y}}(d)+\mathbf{S}(d-7)\right)}$$

where exponentiation is applied elementwise.

$$\tilde{\mathbf{Y}}(d) = \mathbf{S}(d) - \mathbf{S}(d-7)$$

and $\{\mathbf{Y}\}$ is a suitable subset of $\{\tilde{\mathbf{Y}}\}$. The one-day ahead prediction problem then amounts to obtaining the prediction $\hat{\mathbf{L}}(d)$, based on $\{\mathbf{L}(t),\ t < d\}$.

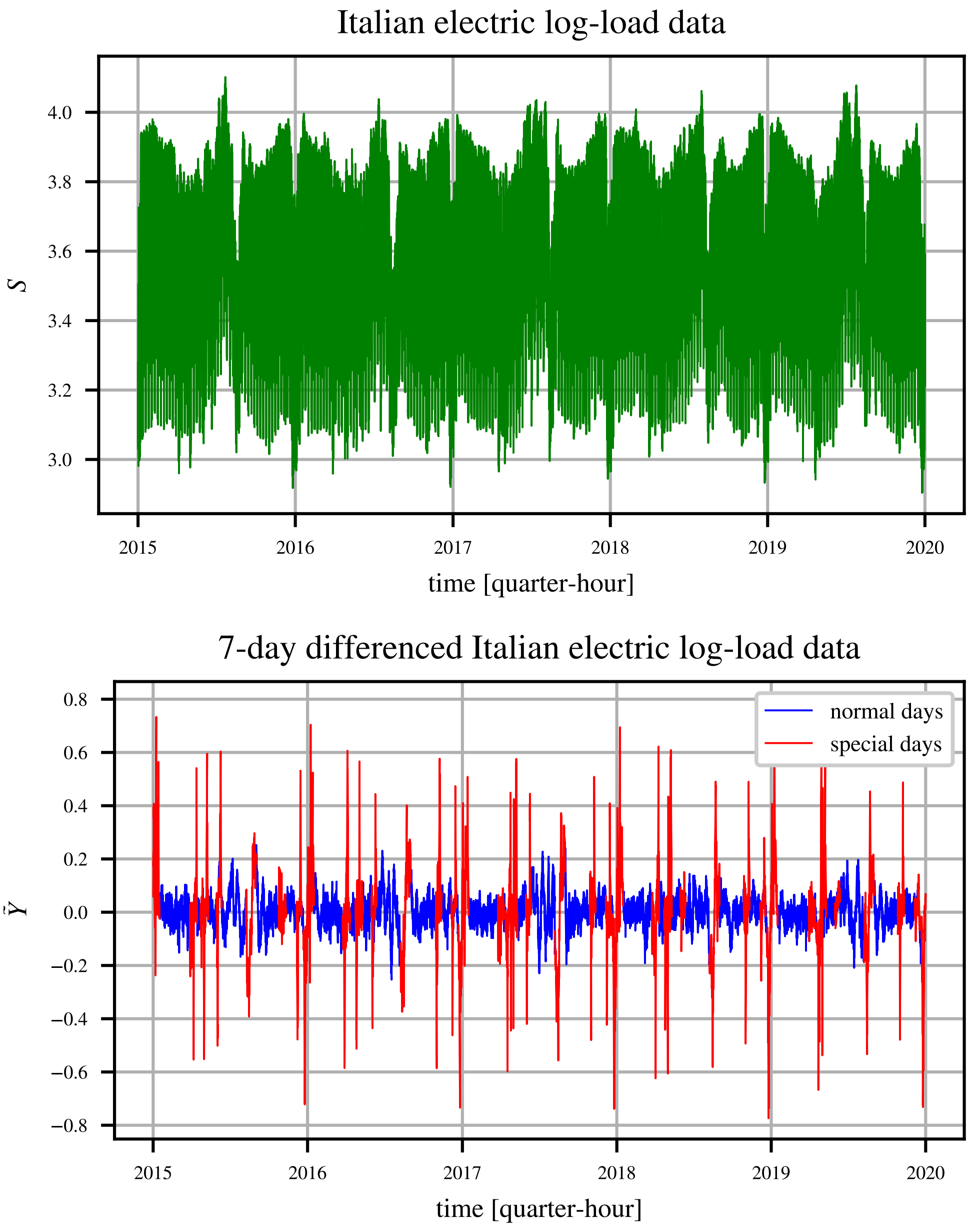


FIGURE 5.3: Preprocessed Italian quarter-hourly electric load demand: log-transformed time series S (top) and 7-day difference of log-transformed time series $\tilde{Y}$ (bottom); data observed in special days are highlighted in red.

The solution approach will go through the calculation of a predictor $\hat{\mathbf{Y}}(d)$ of $\mathbf{Y}(d)$, given $\{\mathbf{Y}(t),\ t < d\}$. Then, the predicted load is straightforwardly obtained as

$$\hat{\mathbf{L}}(d) = e^{\hat{\mathbf{S}}(d)} = e^{\hat{\mathbf{Y}}(d) + \mathbf{S}(d-7)}$$

A general (nonlinear) prediction model can be written as

$$\hat{\mathbf{Y}}(d) = f\left(\mathbf{Y}(d-1), \mathbf{Y}(d-2), \ldots\right)$$

where $f(\cdot,\cdot,\ldots)$ is a suitable nonlinear function. The predictor that minimizes the mean square error

$$\mathrm{MSE} = E\left[\left(\hat{\mathbf{Y}}(d) - \mathbf{Y}(d)\right)^2 \middle| \mathbf{Y}(d-1), \mathbf{Y}(d-2), \ldots\right]$$

is the conditional expectation

$$f\left(\mathbf{Y}(d-1), \mathbf{Y}(d-2), \ldots\right) = E\left[\mathbf{Y}(d) | \mathbf{Y}(d-1), \mathbf{Y}(d-2), \ldots\right]$$

Estimating the conditional expectation is generally a demanding task, but a dramatic simplification occurs when the stochastic process $\{\mathbf{Y}(\cdot)\}$ is Gaussian, in which case the conditional expectation is a linear function of past observations:

$$E\left[\mathbf{Y}(d) | \mathbf{Y}(d-1), \ldots, \mathbf{Y}(d-n)\right] = \sum_{i=1}^{n} \mathbf{A}_i \mathbf{Y}(d-i) \tag{5.1}$$

where $\mathbf{A}_i$ are suitable matrices depending on the second order statistics of the process $\mathbf{Y}(\cdot)$.

Assessing the Gaussianity of a stochastic process is difficult as it involves the Gaussianity of joint distributions of any order. In the present case, we just considered the marginal distributions of the 96 scalar entries of $\mathbf{Y}(\cdot)$ that do not deviate too much from the Gaussian shape. This observation motivates the tentative use of the linear predictor (5.1), where the coefficient matrices $\mathbf{A}_i$ will have to be estimated from data. The next step is the choice of the order n of the predictor. In Fig. 5.4 it is possible to see that the autocorrelation of the signal, as a consequence of the 7-day differentiation, converges to zero rather quickly, suggesting that a low-order predictor may suffice.

In particular the most simple model will be the first-order one

$$\hat{\mathbf{Y}}(d) = \mathbf{A}\mathbf{Y}(d-1) \tag{5.2}$$

where $\mathbf{A}$ is a 96 × 96 matrix of weights defined as:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,96} \\ a_{2,1} & a_{2,2} & \iddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{96,1} & \cdots & \cdots & a_{96,96} \end{bmatrix}$$

This predictor structure lends itself to an insightful interpretation. In fact, $a_{i,j}$ is the weight assigned to $\mathbf{Y}(d-1)_j = Y(d-1,j)$ for predicting $\mathbf{Y}(d)_i = Y(d,i)$. With these assumptions, the purpose of the chapter will be the estimation of the weights $\{a_{i,j}\}$.

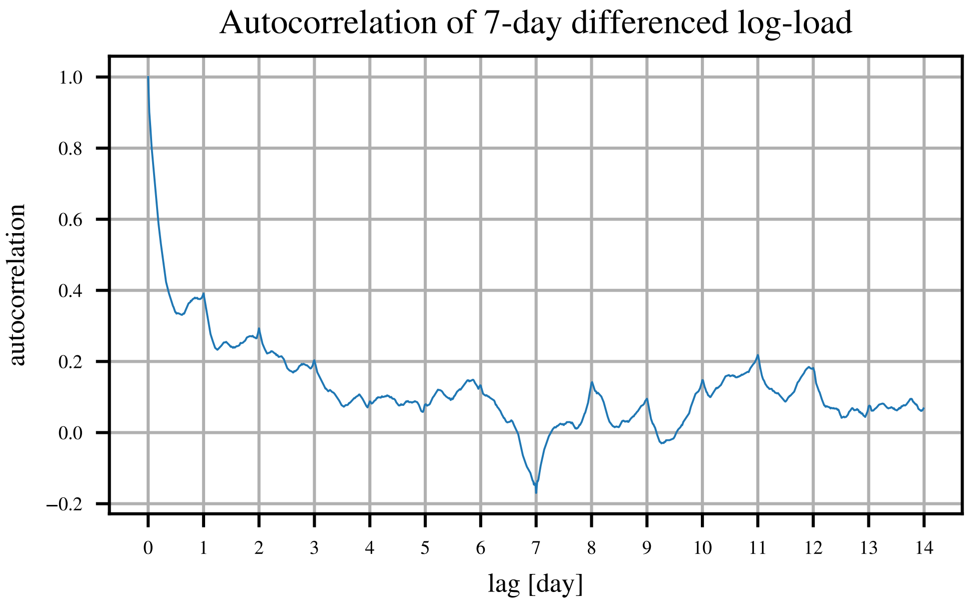


FIGURE 5.4: Autocorrelation of 7-day difference of log-load values computed on the first week of October 2019. As a consequence of the 7-day difference, the autocorrelation tends to stay close to zero as the lag approaches 7 days.

5.3 Forecasting methods

Let

$$\mathbf{a} = \mathrm{vec}(\mathbf{A}) = \left[\begin{array}{ccccccc} a_{1,1} & \dots & a_{1,96} & \dots & a_{96,1} & \dots & a_{96,96} \end{array} \right]^T \in R^{96^2}$$

be the vectorization of $\mathbf{A} \in R^{96\times 96}$. Moreover, let

$$\mathbf{y} = \begin{bmatrix} \mathbf{Y}(1) \\ \mathbf{Y}(2) \\ \vdots \\ \mathbf{Y}(n_{day}) \end{bmatrix} \in R^{96 n_{day}}$$

be the vector of all outputs and

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{I}_{96\times 96} \otimes \mathbf{Y}^T(0) \\ \mathbf{I}_{96\times 96} \otimes \mathbf{Y}^T(1) \\ \vdots \\ \mathbf{I}_{96\times 96} \otimes \mathbf{Y}^T(n_{day}-1) \end{bmatrix} \in R^{96 n_{day} \times 96^2}$$

the regressor matrix, where n_{day} is the number of considered days.

Then, letting $\hat{\mathbf{y}}$ denote the prediction of $\mathbf{y}$, it is possible to rewrite (5.2) as follows:

$$\hat{\mathbf{y}} = \mathbf{\Phi a} \tag{5.3}$$

In the following, six different techniques are considered for the estimation of $\mathbf{a}$: Ordinary Least Squares (LS), a Tikhonov-based amplitude regularization model (TA), a Tikhonov-based second derivative regularization model (TS), a Regularized Radial Basis Functions-based model (RBF), and two sparse models selecting just a suitable subset of the weights, the 'Two-Edges' (TE) and the 'One-Edge' (OnE) models.

5.3.1 Ordinary Least Squares approach (LS)

The most direct way to estimate the weight vector $\mathbf{a}$ is to resort to ordinary least squares:

$$\mathbf{a}^{LS} = \arg\min_{\mathbf{a}} (\mathbf{y} - \mathbf{\Phi a})^T (\mathbf{y} - \mathbf{\Phi a}) \tag{5.4}$$

Recall that each quarter-hour of the target day is predicted as the linear combination of all the quarter-hours of the previous day. Then, it is easy to see that (5.4) is completely equivalent to 96 LS problems, each of which provides the LS estimate of one of the 96 rows of $\mathbf{A}$. This approach is consistent with the multimodel paradigm to the joint design of predictors for different horizons [89], [90]. The multimodel paradigm offers a flexible alternative to the single-model approach that relies on a unique model of a stochastic process from which multistep optimal predictors are computed. However, when a unique model is estimated from data it may suffer from some bias that propagates to the predictors. For instance, if a Prediction Error Method is used for identifying the model, the one-step-ahead predictor errors are minimized, but, if the model is biased, there is no guarantee that long-range predictions are equally satisfactory. Hence the idea of estimating a different predictor for each prediction range, which goes under the name of multi-model approach. In this way, it is possible to reduce the bias of each single predictor, because more degrees of freedom are available. This is obviously more flexible at the cost of possible overparametrization. In our case, in fact, we are estimating $96 \times 96 = 9216$ independent parameters.

In view of the previous considerations, it is not surprising that, when we display as a surface the entries of matrix $\mathbf{A}$ estimated from 2018 data, it turns out to be very rough, see Fig. 5.5. The roughness reflects two features: the variance of the estimates and the oscillations from one column to another. This last feature is best appreciated by looking at the top view displayed in Panel (b) of Fig. 5.5, where the colormap exhibits vertical stripes, a symptom of greater variability across columns than across rows.

This different variability can be explained by considering the problem of predicting the target (i.e. the seven-day difference of the log-loads) at two consecutive quarter-hours i and $i+1$, i.e. the problem of predicting $Y(d,i)$ and $Y(d,i+1)$, given $\mathbf{Y}(d-1)$. The log-loads are sampled frequently and cannot vary abruptly from one quarter-hour to another, a property that propagates to the seven-day difference, so that $Y(d,i) \approx Y(d,i+1)$. Observe also that the two predictors

$$\hat{Y}(d,i) = \sum_{j=1}^{96} \mathbf{a}_{i,j} Y(d-1,j) \tag{5.5}$$

$$\hat{Y}(d,i+1) = \sum_{j=1}^{96} \mathbf{a}_{i+1,j} Y(d-1,j) \tag{5.6}$$

share the same regressors, i.e. the vector $\mathbf{Y}(d-1)$. Since

$$\hat{Y}(d,i) \approx \hat{Y}(d,i+1)$$

it follows that, for each given j, the weights $\mathbf{a}_{i,j}$ and $\mathbf{a}_{i+1,j}$ cannot be too different, which explains the smaller variability across rows.

The irregularity of the surface derives from the overparametrization of the model. On the other hand, there are good reasons for the weight surface to be smooth. Indeed, for any given i, it is reasonable to assume that weights $\mathbf{a}_{i,j}$ and $\mathbf{a}_{i,j+1}$, associated to consecutive quarter-hours, do not differ very much from each other. This justifies the design of alternative estimation schemes that reduce the degrees of freedom by enforcing some kind of smoothness on the weight surface.

5.3.2 Tikhonov regularization

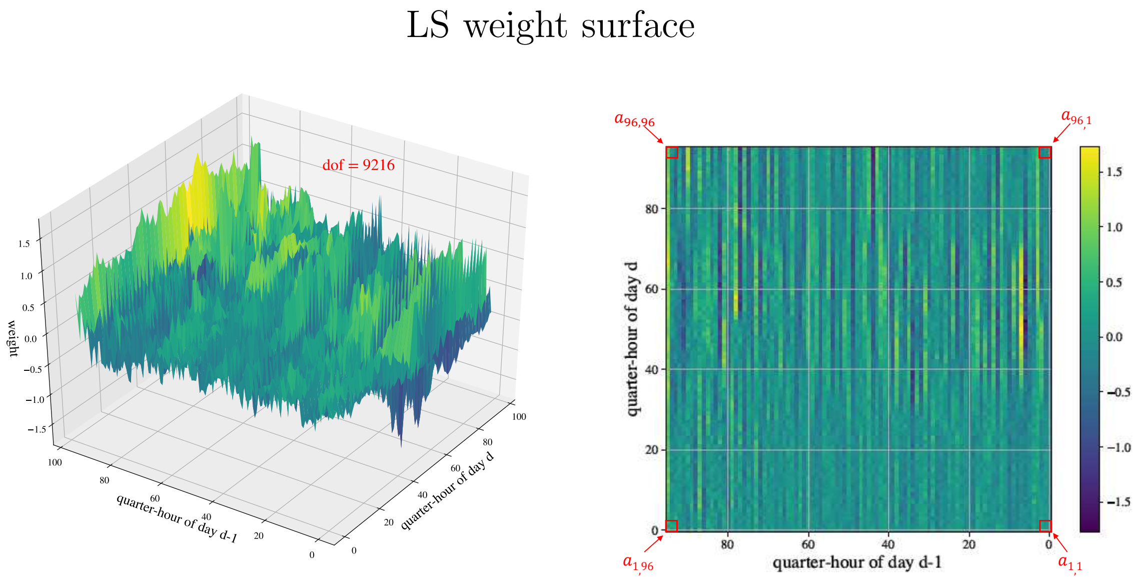
The shortcomings of the LS estimate can be addressed through Tikhonov regularization techniques which, at the cost of some bias, reduce the variance (and the degrees of freedom) of the estimate, by adding a penalty term to the quadratic loss function (5.4) [55], [56]. For the problem of predicting $\mathbf{y}$ by means of $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$, the Tikhonov estimate of $\boldsymbol{\beta}$ is

$$\boldsymbol{\beta}^{reg} = \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \boldsymbol{\beta}^T \mathbf{T} \boldsymbol{\beta} \tag{5.7}$$

where $\mathbf{T} > 0$ is a matrix whose choice determines the type of regularization. For instance, $\mathbf{T} = \lambda \boldsymbol{\Gamma}^T \boldsymbol{\Gamma}$ yields

$$\boldsymbol{\beta}^T \mathbf{T} \boldsymbol{\beta} = \lambda \|\boldsymbol{\Gamma}\boldsymbol{\beta}\|_2^2$$

where $\boldsymbol{\Gamma}$ is the *Tikhonov matrix* and λ is a regularization parameter that controls the balance between the residual sum of squares and the penalty term in (5.7). The tuning of λ can be



(a) 3D view

(b) Top view

FIGURE 5.5: Ordinary Least Squares approach weight surface estimated on the 2018 data: 3D view (left) and top view (right).

performed according to different methods. Hereafter, a cross-validation approach is adopted, whose details are given in Section 5.4.

The solution to (5.7) is

$$\boldsymbol{\beta}^{reg} = \left(\mathbf{X}^T\mathbf{X} + \mathbf{T}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{5.8}$$

so that

$$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}, \quad \mathbf{H} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \mathbf{T}\right)^{-1}\mathbf{X}^T \tag{5.9}$$

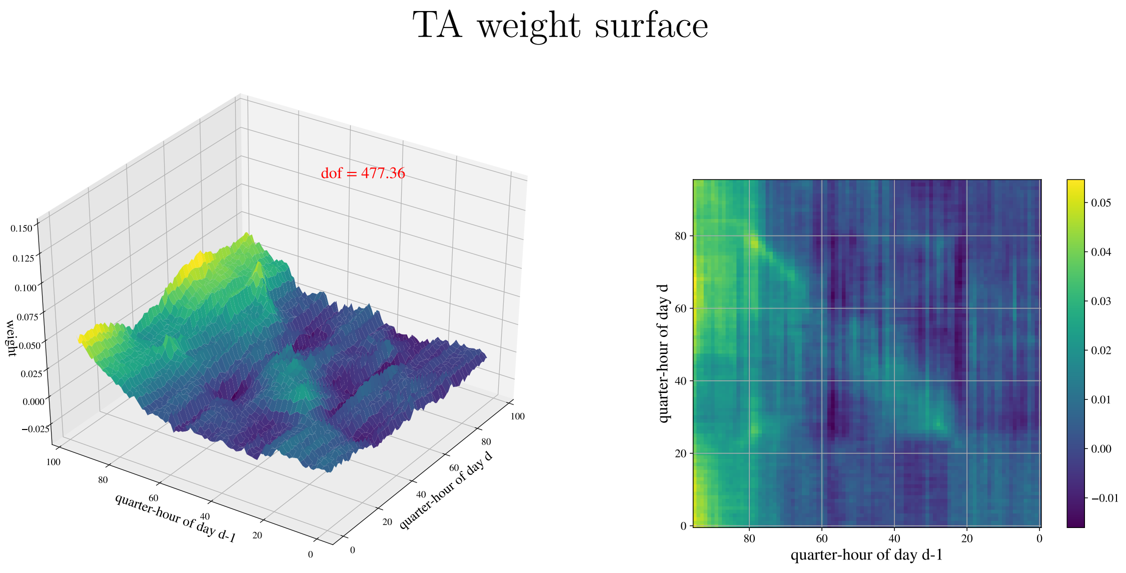
where **H** is the so-called 'hat matrix'. A useful measure of the complexity of the model is given by the equivalent degrees of freedom, that, in linear models, can be computed as the trace of the hat matrix [91]:

$$\mathrm{dof} = \mathrm{Tr}\left(\mathbf{H}\right).$$

Tikhonov amplitude regularization (TA)

A possible regularization strategy consists of applying a penalty to the amplitude of the parameters in order to favor solutions with smaller norm. This technique, also known as *ridge regression*, is associated with the following choice of the **T** matrix:

$$\mathbf{T} = \lambda\mathbf{I}.$$



(a) 3D view

(b) Top view

FIGURE 5.6: Tikhonov amplitude (TA) regularized weight surface estimated on the 2018 data: 3D view (left) and top view (right). In the latter view the yellow and light green colors highlight the presence of two main edges, a vertical and a diagonal one.

Letting $\mathbf{X} = \mathbf{\Phi}$ and $\boldsymbol{\beta} = \mathbf{a}$, the corresponding regularized weight surface estimated from the 2018 data is displayed in Fig. 5.6. Compared to Fig. 5.5, the shape is smoother, although some 'stripe effect' is still visible in Panel (b). Indeed, it is easy to see that solving 5.7 with $\mathbf{T} = \lambda\mathbf{I}$ is equivalent to solving 96 independent ridge regression problems, one for each row of $\mathbf{A}$. In other words, regularity is enforced by damping the amplitude of the entries of $\mathbf{A}$, but oscillations between columns are not explicitly penalized.

It is worth noting that the visual inspection of the Top view (Panel (b) of Fig. 5.6) reveals the presence of two 'edges', one vertical and one diagonal, highlighted by the yellow/light green color. The former is in correspondence with the left side of the square (associated with the last column of matrix $\mathbf{A}$) and the latter is in correspondence of the diagonal of the square (associated with the main diagonal of matrix $\mathbf{A}$). This observation will be the basis of two regularization methods (OnE and TE) that reduce the degrees of freedom by imposing a structured sparse structure on $\mathbf{A}$.

The two edges, that identify the most relevant regressors for the prediction of tomorrow's target variable, admit a very meaningful interpretation. The vertical edge highlights the importance of the most recent observations, i.e. those just before midnight. The diagonal edge, conversely, indicates that, when predicting tomorrow's i-th value $Y(d, i)$, a great weight is assigned to $Y(d-1, i)$, which is rather intuitive.

Tikhonov second derivative regularization (TS)

The idea of this approach is to force the weight surface to be 'smooth' along both directions. In order to do so, two penalty terms are applied to (5.4) in order to penalize the squares of the second differences along both rows and columns. Letting $\mathbf{X} = \mathbf{\Phi}$ and $\boldsymbol{\beta} = \mathbf{a}$, the corresponding Tikhonov regularization problem can be stated as follows:

$$\mathbf{a}^{der2} = \arg\min_{\mathbf{a}} (\mathbf{y} - \mathbf{\Phi a})^T (\mathbf{y} - \mathbf{\Phi a}) + \lambda_1 \|\mathbf{\Delta}_1 \mathbf{a}\|_2^2 + \lambda_2 \|\mathbf{\Delta}_2 \mathbf{a}\|_2^2 \tag{5.10}$$

where λ_1 and λ_2 are two regularization parameters and $\mathbf{\Delta}_1 \in R^{94 \cdot 96 \times 96^2}$ and $\mathbf{\Delta}_2 \in R^{94 \cdot 96 \times 96^2}$ are such that

$$\mathbf{\Delta}_1 \mathbf{a} = \begin{bmatrix} a_{1,3} + 2a_{1,2} - a_{1,1} \\ a_{1,4} + 2a_{1,3} - a_{1,2} \\ \vdots \\ a_{1,96} + 2a_{1,95} - a_{1,94} \\ a_{2,3} + 2a_{2,2} - a_{2,1} \\ \vdots \\ a_{96,96} + 2a_{96,95} - a_{96,94} \end{bmatrix}, \quad \mathbf{\Delta}_2 \mathbf{a} = \begin{bmatrix} a_{3,1} + 2a_{2,1} - a_{1,1} \\ a_{4,1} + 2a_{3,1} - a_{2,1} \\ \vdots \\ a_{96,1} + 2a_{95,1} - a_{94,1} \\ a_{3,2} + 2a_{2,2} - a_{1,2} \\ \vdots \\ a_{96,96} + 2a_{95,96} - a_{94,96} \end{bmatrix}$$

yield the row-wise and column-wise second differences of the entries of $\mathbf{A}$. It is immediate to see that the corresponding $\mathbf{T}$ matrix is

$$\mathbf{T} = \lambda_1 \mathbf{\Delta}_1^T \mathbf{\Delta}_1 + \lambda_2 \mathbf{\Delta}_2^T \mathbf{\Delta}_2.$$

It is worth noting that if $\lambda_1 = \lambda_2$ the regularization penalty in (5.10) boils down to the classical discrete Laplacian operator. In this work, the formulation with two independent regularization parameters is preferred in view of its greater flexibility.

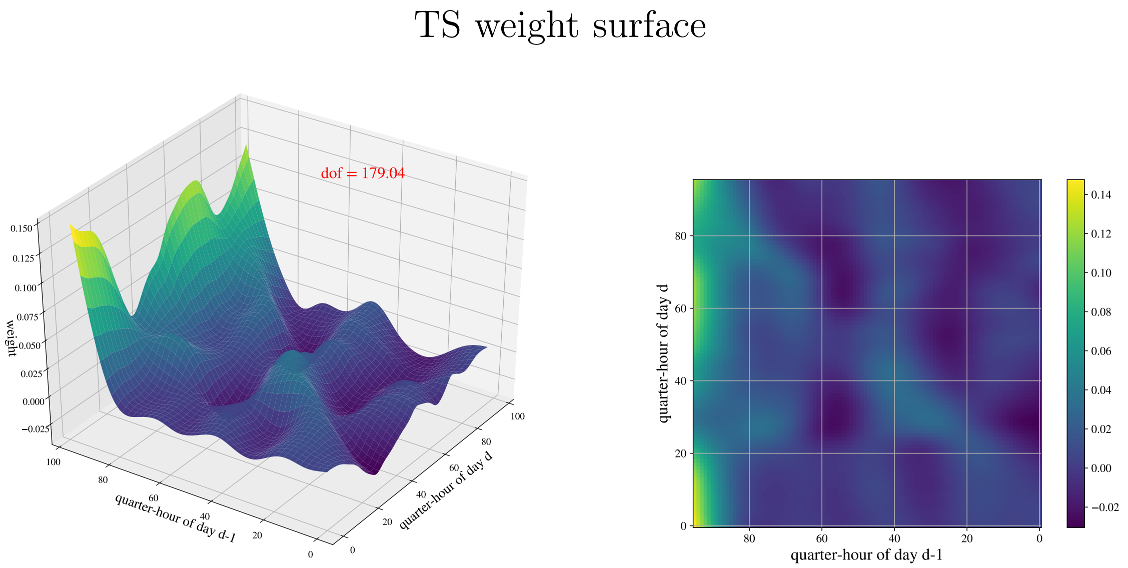
The corresponding surface, displayed in Fig. 5.7, is even smoother than the TA one. The vertical and diagonal edges are still well seen in Panel (b).

5.3.3 Regularized Radial Basis Functions regularization (RBF)

The approach described in this subsection consists of regularizing the weight surface by representing it as the sum of a cubic polynomial term, which is used to capture its trends, and a set of Regularized Radial Basis Functions (RBF) to capture the local details of the surface [92], [93]. In particular:

$$a_{i,j} = \bar{a}_{i,j} + \tilde{a}_{i,j}$$

$$\bar{a}_{i,j} = c_1 + c_2 i + c_3 j + c_4 i^2 + c_5 ij + c_6 j^2 + c_7 i^3 + c_8 i^2 j + c_9 ij^2 + c_{10} j^3$$



(a) 3D view

(b) Top view

FIGURE 5.7: Tikhonov second derivative regularized weight surface estimated on the 2018 data: 3D view (left) and top view (right).

$$\tilde{a}_{i,j} = \sum_{k=0}^{m} \sum_{z=0}^{m} \theta_{k,z} \phi \left(\sqrt{(i - w_k)^2 + (j - v_z)^2} \right)$$

where

$$\phi(r) = e^{-\frac{r^2}{2\sigma^2}}$$

and (w_k, v_z), with $k, z = 1, ..., m$, are the coordinates of the centers of the radial functions that are assumed to be located on a uniform square grid:

$$w_k = \frac{96k}{m}, \quad k = 0, \ldots, m \tag{5.11}$$

$$w_z = \frac{96z}{m}, \quad z = 0, \ldots, m \tag{5.12}$$

and σ is the standard deviation.

Notice that the parameter vector $\mathbf{a}$ can be written as

$$\mathbf{a} = \bar{\mathbf{a}} + \tilde{\mathbf{a}}$$

with the corresponding weight surface given by

$$\mathbf{A} = \bar{\mathbf{A}} + \tilde{\mathbf{A}}$$

where $\bar{\mathbf{A}}$ denotes the polynomial component and $\tilde{\mathbf{A}}$ the RBF one. Once again, the optimization problem is the one given in (5.7), with

$$\mathbf{y} = \boldsymbol{\Phi}(\bar{\mathbf{a}} + \tilde{\mathbf{a}}), \quad \bar{\mathbf{a}} = \mathbf{P}\mathbf{c}, \quad \tilde{\mathbf{a}} = \mathbf{R}\boldsymbol{\theta}$$

so that

$$\mathbf{X} = \boldsymbol{\Phi}\left[\begin{array}{cc} \mathbf{P} & \mathbf{R} \end{array}\right], \quad \boldsymbol{\beta} = \left[\begin{array}{c} \mathbf{c} \\ \boldsymbol{\theta} \end{array}\right]$$

where the matrices $\mathbf{P}$ and $\mathbf{R}$ are the cubic polynomial and the radial basis functions matrices, respectively. The matrix $\mathbf{T}$ applies ridge regularization to the amplitudes of the radial basis functions, while no shrinking is applied to the polynomial coefficients.

$$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \lambda\mathbf{I_m} \end{bmatrix}$$

In particular,

$$\mathbf{P} = \left[\begin{array}{cccccc} \mathbf{1} & \mathbf{i} & \mathbf{j} & \mathbf{i^2} & \dots & \mathbf{j^3} \end{array}\right] \in R^{96^2 \times 10}$$

$$\mathbf{R} = \begin{bmatrix} \phi\left(\sqrt{(1-w_0)^2 + (1-v_0)^2}\right) & \dots & \phi\left(\sqrt{(1-w_m)^2 + (1-v_m)^2}\right) \\ \vdots & \vdots & \vdots \\ \phi\left(\sqrt{(1-w_0)^2 + (96-v_0)^2}\right) & \dots & \phi\left(\sqrt{(1-w_m)^2 + (96-v_m)^2}\right) \\ \\ \phi\left(\sqrt{(2-w_0)^2 + (1-v_0)^2}\right) & \dots & \phi\left(\sqrt{(2-w_m)^2 + (1-v_m)^2}\right) \\ \vdots & \vdots & \vdots \\ \phi\left(\sqrt{(2-w_0)^2 + (96-v_0)^2}\right) & \dots & \phi\left(\sqrt{(2-w_m)^2 + (96-v_m)^2}\right) \\ \vdots & \vdots & \vdots \\ \phi\left(\sqrt{(96-w_0)^2 + (1-v_0)^2}\right) & \dots & \phi\left(\sqrt{(96-w_m)^2 + (1-v_m)^2}\right) \\ \vdots & \vdots & \vdots \\ \phi\left(\sqrt{(96-w_0)^2 + (96-v_0)^2}\right) & \dots & \phi\left(\sqrt{(96-w_m)^2 + (96-v_m)^2}\right) \end{bmatrix} \in R^{96^2 \times m^2}$$

where

$$\mathbf{i} = \left[\begin{array}{cccccccccc} 1 & \dots & 96 & 1 & \dots & 96 & \dots & 1 & \dots & 96 \end{array}\right]^T \in R^{96^2 \times 1},$$

$$\mathbf{j} = \left[\begin{array}{cccccccccc} 1 & \dots & 1 & 2 & \dots & 2 & \dots & 96 & \dots & 96 \end{array}\right]^T \in R^{96^2 \times 1}$$

$$\mathbf{c} = \left[\begin{array}{cccc} c_1 & c_2 & \dots & c_{10} \end{array}\right]^T \in R^{10 \times 1}$$

$$\boldsymbol{\theta} = \left[\begin{array}{cccccc} \theta_{1,1} & \dots & \theta_{1,m} & \theta_{2,1} & \dots & \theta_{m,m} \end{array} \right]^T \in R^{m^2 \times 1}$$

In this chapter, the value of σ has been fixed to 4 while m has been fixed to 12 (which leads to $13 \times 13 = 169$ bell-shaped basis functions). The vector θ contains the parameters $\theta_{k,z}$, each one representing the amplitude of the radial function centered in (w_k, v_z).

The cubic surface component $\bar{\mathbf{A}}$, the regularized radial basis function surface $\tilde{\mathbf{A}}$ and the final surface $\mathbf{A}$ are shown in Fig. 5.8(a), 5.8(b) and 5.8(c), respectively.

Again, the visual inspection of the Top view (Panel (d) of Fig. 5.8) reveals the presence of two 'edges' a horizontal and a diagonal, highlighted by the yellow/light green color. The dominance of such edges motivates the exploration of the sparse identification strategy, described in the next subsection.

5.3.4 Two-edges model (TE)

We now consider a simplified sparse model of the weight surface $\mathbf{A}$. The name 'Two-edges' is due to the fact that only two vectors (herein called 'edges') are estimated instead of a full 96×96 surface of weights, namely the last column

$$\mathbf{a}^{last} = \left[\begin{array}{cccc} a_{1,96} & a_{2,96} & \dots & a_{95,96} \end{array} \right]^T \in R^{95 \times 1}$$

and the main diagonal

$$\mathbf{a}^{diag} = \left[\begin{array}{cccc} a_{1,1} & a_{2,2} & \dots & a_{96,96} \end{array} \right]^T \in R^{96 \times 1}$$

of $\mathbf{A}$, which leads to a model with 191 parameters (note that $a_{96,96}$ is shared by the last column and the main diagonal).

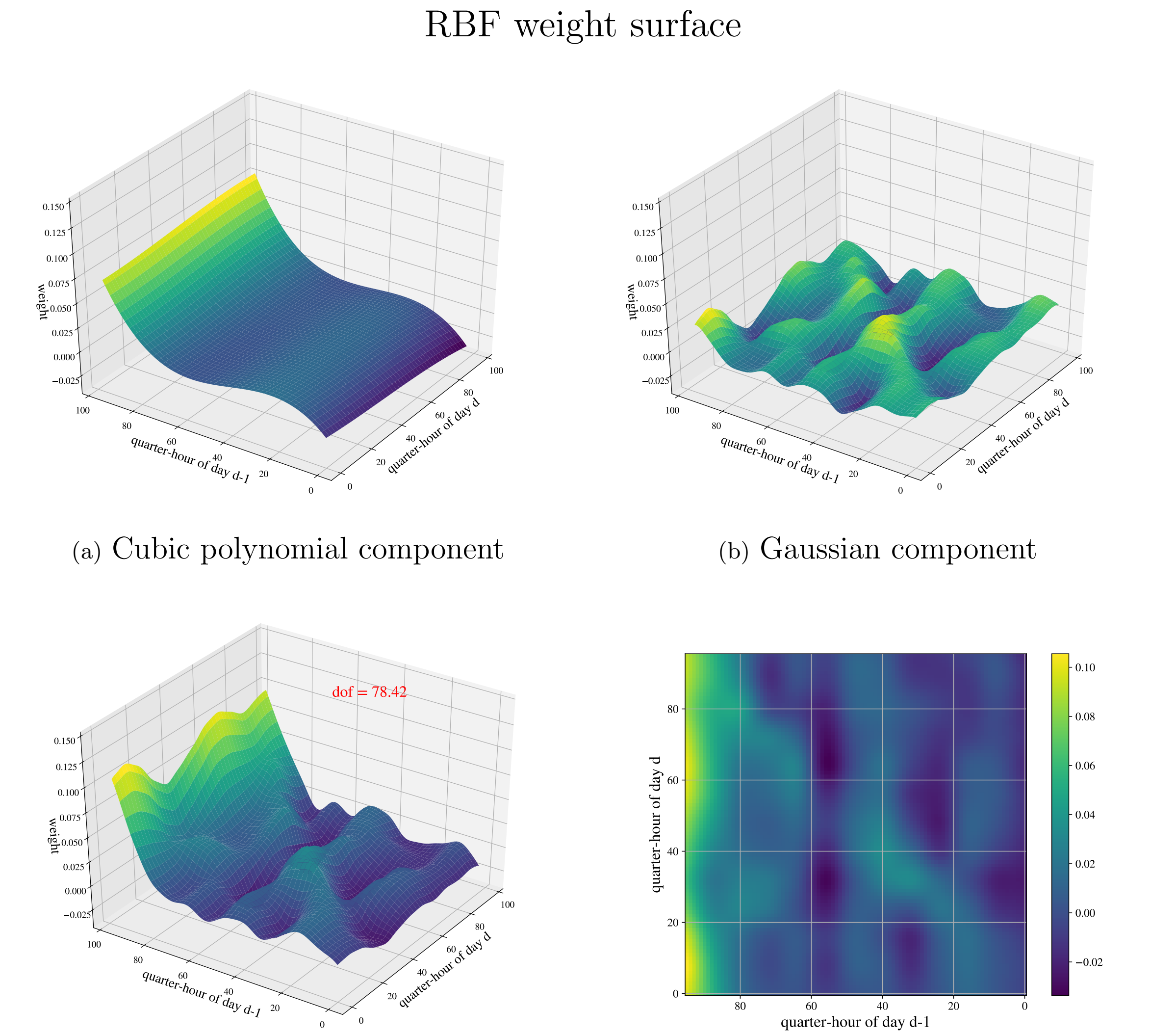
The new model formulation is

$$\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$$

with $\boldsymbol{\beta}\left(\mathbf{a}^{diag}, \mathbf{a}^{last}\right) = \left[\begin{array}{cccccc} a_{1,1} & a_{1,96} & a_{2,2} & a_{2,96} & \dots & a_{96,96} \end{array} \right]^T \in R^{191 \times 1}$ and

$$\boldsymbol{\Psi} = \left[\begin{array}{c} \mathbf{I}_{96 \times 96} \otimes \boldsymbol{\psi}_k^T(0) \\ \mathbf{I}_{96 \times 96} \otimes \boldsymbol{\psi}_k^T(1) \\ \vdots \\ \mathbf{I}_{96 \times 96} \otimes \boldsymbol{\psi}_k^T(n_{day} - 1) \end{array} \right]^T \in R^{96 n_{day} \times 1}$$

with $\boldsymbol{\psi}_k(j) = \left[\begin{array}{cc} y(j,k) & y(j,96) \end{array} \right]^T$.



(a) Cubic polynomial component

(b) Gaussian component

(c) Regularized Radial Basis Functions surface (3D view)

(d) Regularized Radial Basis Functions surface (Top view)

FIGURE 5.8: Regularized Radial Basis Functions weight surface estimated on the 2018 data: cubic polynomial component (top left), Gaussian component (top right), final Regularized Radial Basis Functions surface 3D view (bottom left) and top view (bottom right).

Further regularization can be introduced by adding two penalty terms λ_{diag} and λ_{last} that shrink the second derivatives of each edge. The optimization problem becomes

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\beta})^T (\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\beta}) + \lambda_{last} \|\boldsymbol{\Delta}_{last}\boldsymbol{\beta}\|_2^2 + \lambda_{diag} \|\boldsymbol{\Delta}_{diag}\boldsymbol{\beta}\|_2^2$$

where $\boldsymbol{\Delta}_{last} \in R^{94\times 191}$ and $\boldsymbol{\Delta}_{diag} \in R^{94\times 191}$ are such that

$$\boldsymbol{\Delta}_{last}\boldsymbol{\beta} = \begin{bmatrix} a_{3,96} - 2a_{2,96} + a_{1,96} \\ a_{4,96} - 2a_{3,96} + a_{2,96} \\ \vdots \\ a_{96,96} - 2a_{95,96} + a_{94,96} \end{bmatrix} \tag{5.13}$$

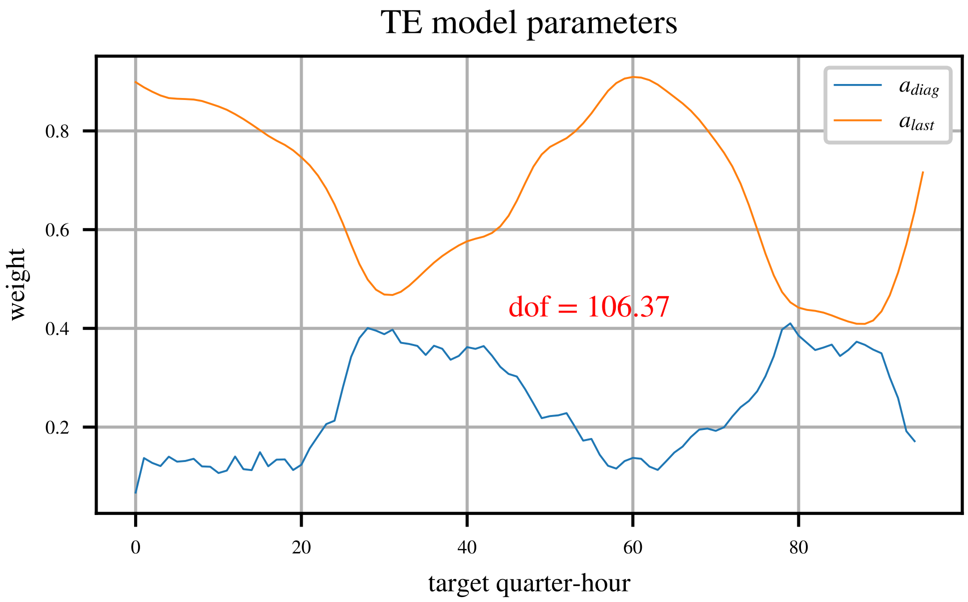


FIGURE 5.9: Behaviour of the parameters of the TE model as a function of the quarter-hour of the target value.

$$\mathbf{\Delta}_{diag}\boldsymbol{\beta} = \begin{bmatrix} a_{3,3} - 2a_{2,2} + a_{1,1} \\ a_{4,4} - 2a_{3,3} + a_{2,2} \\ \vdots \\ a_{96,96} - 2a_{95,95} + a_{94,94} \end{bmatrix} \tag{5.14}$$

The corresponding $\mathbf{T}$ matrix is

$$\mathbf{T} = \lambda_{last}\mathbf{\Delta}_{last}^{T}\mathbf{\Delta}_{last} + \lambda_{diag}\mathbf{\Delta}_{diag}^{T}\mathbf{\Delta}_{diag}.$$

The estimated parameters $\mathbf{a}^{last}$ and $\mathbf{a}^{diag}$ are shown in Fig. 5.9. It is clear that $\mathbf{a}^{diag}$ and $\mathbf{a}^{last}$ are correlated, which suggests a further simplification of the parametrization, which is discussed in the next subsection.

5.3.5 One-edge model (OnE)

As a consequence of the correlation between $\mathbf{a}^{last}$ and $\mathbf{a}^{diag}$, it is worth to try to develop an even simpler model which consider just one between $\mathbf{a}^{last}$ and $\mathbf{a}^{diag}$. Based on intuition, it makes more sense to consider the latter one over the former one. Therefore the ‘One-edge’ model reduces the weight matrix $\mathbf{A}$ just to its diagonal entries, that is the vector of parameters a^{diag}. According to this model, the 7-day difference at a certain quarter-hour of the target day is proportional to the 7-day difference of the previous day at the same quarter-hour.

In the resulting model, $\boldsymbol{\beta} = \mathbf{a}_{diag}$ and $\mathbf{X} = \boldsymbol{\Xi}$ with

$$\boldsymbol{\Xi} = \begin{bmatrix} diag\left(\mathbf{Y}(0)\right) \\ diag\left(\mathbf{Y}(1)\right) \\ \vdots \\ diag\left(\mathbf{Y}(n_{day}-1)\right) \end{bmatrix}^T \in R^{96 n_{day} \times 96}$$

where the diagonalization operator $diag(\cdot)$ is defined as

$$diag\left(\mathbf{Y}(d)\right) = \begin{bmatrix} y_{d,1} & 0 & 0 & \dots & 0 \\ 0 & y_{d,2} & 0 & \dots & 0 \\ 0 & 0 & y_{d,3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & y_{d,96} \end{bmatrix} \in R^{96 \times 96}$$

A penalty term on the second derivative of the parameter vector is included in the cost function:

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} (\mathbf{y} - \boldsymbol{\Xi}\boldsymbol{\beta})^T (\mathbf{y} - \boldsymbol{\Xi}\boldsymbol{\beta}) + \lambda_{diag} \|\boldsymbol{\Delta}_{diag}\boldsymbol{\beta}\|_2^2$$

with $\boldsymbol{\Delta}_{diag}$ defined as in (5.14). The corresponding $\mathbf{T}$ matrix is given by

$$\mathbf{T} = \lambda_{diag} \boldsymbol{\Delta}_{diag}^T \boldsymbol{\Delta}_{diag}.$$

The regularized solution, shown in Fig. 5.10, is characterized by a dramatic decrease of the degrees of freedom, compared to the other models. Before illustrating the results of the prediction models, that will be presented in Section 5.5, we introduce another methodology relying on an aggregation paradigm.

5.3.6 Aggregated forecast

The combination of multiple forecasts is a commonly adopted technique in order to improve forecasts [2]. A great variety of aggregation techniques has been proposed in the literature [9], [14], [15], [16]. The simple average method is in general an effective and robust strategy adopted to reduce the prediction variance [14], [15], [16].

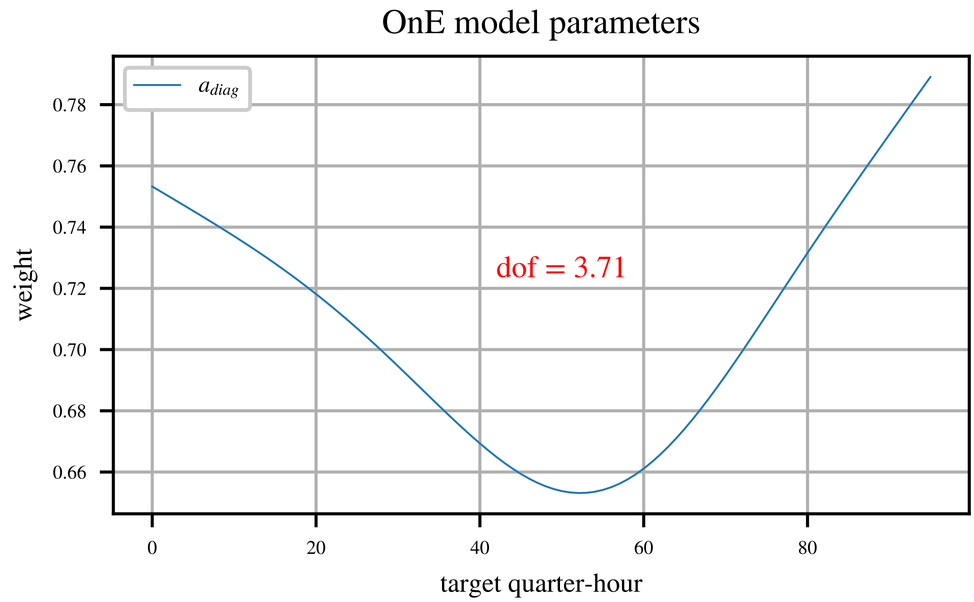


FIGURE 5.10: One-edge model parameters behaviour as a function of the quarter-hour of the target value.

Let $\mathcal{M}_i, i = 1, \ldots, m$, denote a set of prediction models and $\hat{L}_{\mathcal{M}_i}(d, q)$ the corresponding load demand forecasts generated by $\mathcal{M}_i$. Then

$$\hat{L}_{\text{Avg}}(d, q) = \frac{1}{m} \sum_{i=1}^{m} \hat{L}_{\mathcal{M}_i}(d, q)$$

is the aggregated forecast obtained by averaging the predictions generated by $\mathcal{M}_i$.

In order to better understand the potential benefit of aggregation, consider the simple case of just two prediction models, i.e. $m = 2$, and define the corresponding residuals as:

$$e_{\mathcal{M}_i}(d, q) = L(d, q) - \hat{L}_{\mathcal{M}_i}(d, q)$$

It is immediate to verify that

$$e_{\text{Avg}}(d, q) = L(d, q) - \hat{L}_{\text{Avg}}(d, q) = \frac{1}{2} \left(e_{\mathcal{M}_1}(d, q) + e_{\mathcal{M}_2}(d, q) \right)$$

Then the Mean Squared Error (MSE) of the aggregated forecast is given by:

$$\begin{aligned} MSE_{\text{Avg}} &= E\left[e_{\text{Avg}}^2\right] = E\left[\frac{1}{4}\left(e_{\mathcal{M}_1}^2 + e_{\mathcal{M}_2}^2 + 2 e_{\mathcal{M}_1} e_{\mathcal{M}_2}\right)\right] = \\ &= \frac{1}{4}\left(E\left[e_{\mathcal{M}_1}^2\right] + E\left[e_{\mathcal{M}_2}^2\right] + 2E\left[e_{\mathcal{M}_1} e_{\mathcal{M}_2}\right]\right) = \\ &= \frac{1}{4} MSE_{\mathcal{M}_1} + \frac{1}{4} MSE_{\mathcal{M}_2} + \frac{1}{2}\left(Cov\left[e_{\mathcal{M}_1} e_{\mathcal{M}_2}\right] + E\left[e_{\mathcal{M}_1}\right] E\left[e_{\mathcal{M}_2}\right]\right) = \\ &= \frac{1}{4} MSE_{\mathcal{M}_1} + \frac{1}{4} MSE_{\mathcal{M}_2} + \frac{1}{2}\left(\rho_{e_{\mathcal{M}_1}, e_{\mathcal{M}_2}} \sigma_{e_{\mathcal{M}_1}} \sigma_{e_{\mathcal{M}_2}} + E\left[e_{\mathcal{M}_1}\right] E\left[e_{\mathcal{M}_2}\right]\right) \end{aligned} \tag{5.15}$$

where $\rho_{e_{\mathcal{M}_1},e_{\mathcal{M}_2}}$ is the coefficient of correlation between $e_{\mathcal{M}_1}$ and $e_{\mathcal{M}_2}$.

Assume that $\mathcal{M}_1$ performs better than $\mathcal{M}_2$, that is $MSE_{\mathcal{M}_1} < MSE_{\mathcal{M}_2}$. Then the aggregated forecaster $\hat{L}_{\text{Avg}}(d,q)$ improves on $\hat{L}_{\mathcal{M}_1}(d,q)$ provided that the following inequality is satisfied:

$$MSE_{\mathcal{M}_2} < 3MSE_{\mathcal{M}_1} - 2E[e_{\mathcal{M}_1}e_{\mathcal{M}_2}] - 2E\left[e_{\mathcal{M}_1}\right]E\left[e_{\mathcal{M}_2}\right]$$

where $E\left[e_{\mathcal{M}_1}\right]$ and $E\left[e_{\mathcal{M}_2}\right]$ are the bias of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively.

Consider for instance the case when at least one of the predictors has a negligible bias. Then, provided that the covariance between the errors $e_{\mathcal{M}_1}$ and $e_{\mathcal{M}_2}$ is small, the aggregation between the two predictors brings an improvement even when the MSE of the worse performing one is up to three times larger than that of the best performing one. As a consequence, there is room for designing predictors that employ different strategies aiming at obtaining scarcely correlated prediction errors. Later on, the correlation between the residuals of the newly proposed method and those of the Italian TSO predictor will be studied in order to assess the potential benefits ensuing from an aggregation.

5.4 Experimental validation setup

Three scenarios were considered for evaluating and comparing the proposed models:

- Training = 2016, Test = 2017
- Training = 2017, Test = 2018
- Training = 2018, Test = 2019

This choice is driven by the fact that the Terna forecasts, that are used as benchmark for comparison purposes, are available for the years 2017, 2018, 2019.

The performances of the models were evaluated using three metrics: $MAPE$, $RMSE$ and MAE, each one both on quarter-hourly and daily data, for a total of six performance indexes:

$$MAPE = \frac{100}{n} \sum_{d \in \mathcal{D}_{Te}} \sum_{q=1}^{96} \left| \frac{L(d,q) - \hat{L}(d,q)}{L(d,q)} \right|$$

$$RMSE = \sqrt{\frac{\sum_{d \in \mathcal{D}_{Te}} \sum_{q=1}^{96} \left(L(d,q) - \hat{L}(d,q) \right)^2}{n}}$$

$$MAE = \frac{\sum_{d \in \mathcal{D}_{Te}} \sum_{q=1}^{96} \left| L(d,q) - \hat{L}(d,q) \right|}{n}$$

$$MAPE_{daily} = \frac{100}{n_{day}} \sum_{d \in \mathcal{D}_{Te}} \left| \frac{L_{daily}(d) - \hat{L}_{daily}(d)}{L_{daily}(d)} \right|$$

$$RMSE_{daily} = \sqrt{\frac{\sum_{d \in \mathcal{D}_{Te}} \left(L_{daily}(d) - \hat{L}_{daily}(d) \right)^2}{n_{day}}}$$

$$MAE_{daily} = \frac{\sum_{d \in \mathcal{D}_{Te}} \left| L_{daily}(d) - \hat{L}_{daily}(d) \right|}{n_{day}}$$

where $\mathcal{D}_{Te}$ is the set of test days, n is the number of quarter-hourly test data, and n_{day} is the number of daily test data. In particular, $\mathcal{D}_{Te} = \{d : d \notin \mathcal{D}_s, d - 7 \notin \mathcal{D}_s\}$, where $\mathcal{D}_s$ includes special days such as Winter, Summer, Easter and national holidays (see Appendix A). L_{daily} and $\hat{L}_{daily}$ are respectively the time series of daily averages of electric load observations and the associated forecasts:

$$L_{daily}(d) = \frac{1}{96} \sum_{q=1}^{96} L(d,q), \quad \hat{L}_{daily}(d) = \frac{1}{96} \sum_{q=1}^{96} \hat{L}(d,q)$$

The hyperparameters of the models described in Section 5.3 are tuned through cross-validation choosing the $MAPE$ as objective function and using, for each scenario, the two years preceding the test one as training and validation: e.g. for the first scenario (test year = 2017) cross-validation is performed on the years 2015 and 2016, using the first year for training and the second one for validation.

All the results of the hyperparameters tuning phase are summarized in Table 5.1.

TABLE 5.1: Hyperparameters tuning: cross-validation results

	TA	**TS**	**RBF**	**TE**	**OnE**
Train: 2015 **Val: 2016**	$\lambda = 0.1$	$\lambda_1 = 10$ $\lambda_2 = 100$	$\lambda = 1$	$\lambda_{diag} = 0.01$ $\lambda_{last} = 100$	$\lambda_{diag} = 100$
Train: 2016 **Val: 2017**	$\lambda = 0.1$	$\lambda_1 = 10$ $\lambda_2 = 1$	$\lambda = 10$	$\lambda_{diag} = 1$ $\lambda_{last} = 0.01$	$\lambda_{diag} = 10000$
Train: 2017 **Val: 2018**	$\lambda = 1$	$\lambda_1 = 100$ $\lambda_2 = 10$	$\lambda = 10$	$\lambda_{diag} = 0.01$ $\lambda_{last} = 1$	$\lambda_{diag} = 10000$

5.5 Forecasting results

In this section, the predictive performances of the models described in Section 5.3 are discussed and compared to the Terna forecaster. The results for the three test scenarios are summarized in Table 5.3, 5.4, 5.5, where, for each predictor, the performance indexes introduced in Section 5.4 and the Degrees of Freedom (dof), accounting for the complexity of the underlying models, are reported.

In all scenarios the LS approach performs poorly: it achieves the worst performances and has too many degrees of freedom (96 x 96 = 9216 = # of parameters).

On the other hand, the OnE model is too parsimonious: while it achieves results that improve on the LS approach and in some cases are comparable to the benchmark ones (e.g. see 2018 and 2019 $MAPE$), its performances are significantly worse on the 2017 scenario and in all daily indexes.

Significant improvements are obtained by resorting to regularized predictors. The TA, TS and RBF models achieve comparable results over the three test years, the main difference being represented by the complexity of the three approaches, where the last predictor stands out for being the most parsimonious one (dof around 178 in the three scenarios).

The TE model, which further reduces the complexity of the predictor (dof in the range from 100 to 115) ranks first in the 2018 scenario, while it is slightly inferior to the RBF predictor in the other two scenarios. In view of this, it provides an effective compromise between accuracy and simplicity.

The percentage decreases of $MAPE$ and the MAE brought by TA, TS, RBF and TE with respect to the benchmark reach 20% in 2018 and 24% in 2019, while they are less evident in terms of $RMSE$ (2% in 2018, 12% in 2019). This can be explained by a better accuracy of the proposed predictors during the night (where there are lower demands) and by the presence of few large errors within the predictions (to which the $RMSE$ score is more sensitive than $MAPE$ and MAE).

The 2017 scenario is the tougher one. In particular, while percentage improvements on the quarter-hourly $MAPE$ and MAE are relatively small (7%), the quarter-hourly $RMSE$ results achieved by Terna are slightly better than ours. Moreover, the daily performances of Terna are superior than the ones achieved by the proposed models.

This is the consequence of a few large errors within the proposed forecasts, possibly related to the fact that the proposed predictors do not account for any exogenous variables such as the temperature, which can be crucial for the prediction in some seasons of the year and some phases of the day. By contrast, this information is exploited by the Terna forecaster (see Appendix B).

In Fig. 5.11, 5.12 and 5.13 it is possible to visualize how the forecasts of the RBF and TE models compare to the Terna ones over different weeks of 2017, 2018 and 2019. In particular, the residual plots reveal that, while the error profiles of the proposed models are similar, they are almost uncorrelated with the Terna forecast error, as confirmed by the inspection of the scatter plots in 2017, 2018 and 2019 of the residuals for Terna vs RBF (correlation coefficient $\rho_{e_{\mathcal{M}_{RBF}}, e_{\mathcal{M}_{Terna}}}$ ranging from 0.26 to 0.31) and Terna vs TE ($\rho_{e_{\mathcal{M}_{TE}}, e_{\mathcal{M}_{Terna}}}$ ranging 0.3 from to 0.35), see Fig. 5.14. It turns out that the absolute values of the biases of RBF and TE are always less than 0.1. In view of this, there is room for improving the quality of the forecasts by combining Terna's predictions with those produced by the new proposed methods [14], [16]. The margin for improvement was assessed by plugging into formula (5.15) the values reported in Table 5.2. The formula predicts that a significant improvement can be achieved. In particular, the best MSE that ranges from 0.98 (RBF) in 2019 to 1.13 $\left[GW^2\right]$ (Terna) in 2017 is predicted to range between 0.71 and 0.81 throughout the considered years when the aggregated predictors AVG(RBF) and AVG(TE) are employed.

All the aggregated predictions obtained from the models proposed in Section 5.3 were evaluated according to the same framework. The results, summarized in Tables 5.6, 5.7, 5.8, highlight a very substantial improvement with respect to Terna benchmark in all three scenarios, for all performances indexes and in both the quarter-hourly and the daily cases. In particular, the improvement with respect to the TSO benchmark for the quarter-hourly indexes is always not less than 20%, reaching 32% for the $MAPE$ 2019 (Avg(RBF)), while the improvement of the daily indexes is always not less than 15%, reaching 35% for $MAPE_{daily}$ and MAE_{daily} in 2019 (Avg(RBF)). The predicted performances of the aggregated forecasters provided by formula (5.15) are remarkably accurate: the error is always not greater than 0.02 $\left[GW^2\right]$, see Table 5.2.

The time plots and residual plots for the Avg(RBF) and the Avg(TE) cases over some sample weeks of 2017,2018 and 2019 are displayed in Fig. 5.15, 5.16 and 5.17.

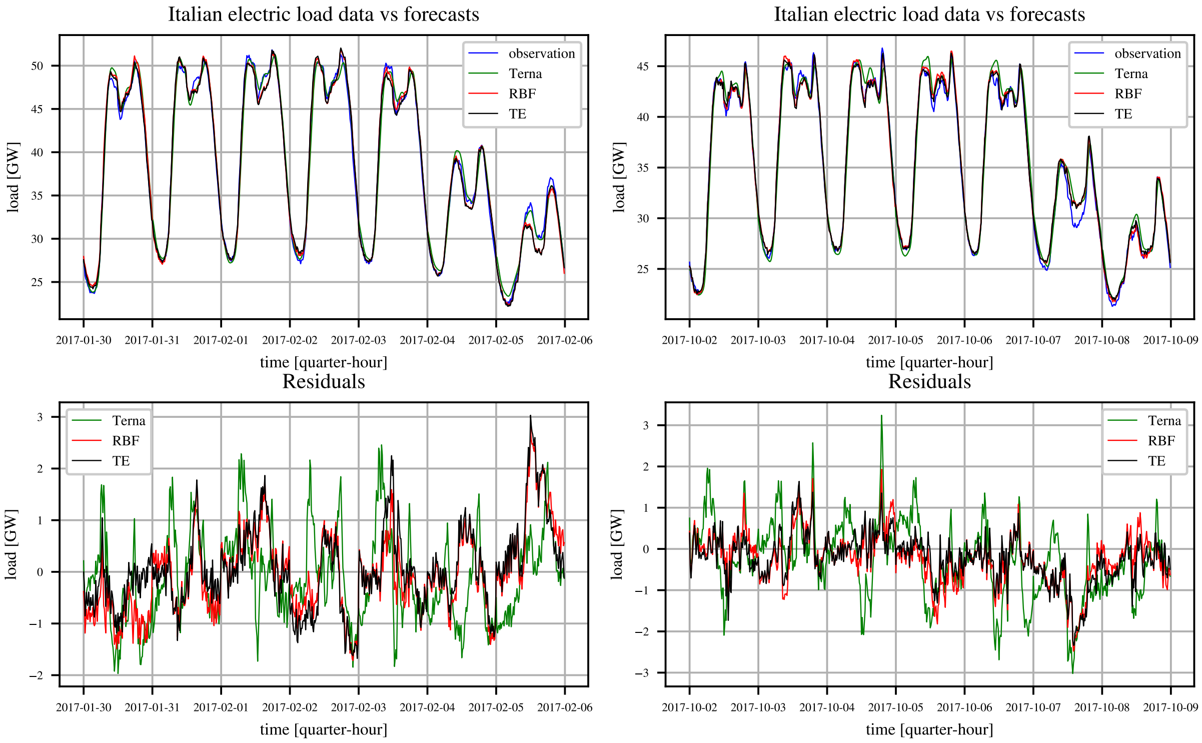


FIGURE 5.11: RBF, TE and Terna prediction (top) and residual (bottom) over two sample weeks on 2017.

Overall, the new prediction strategy offers the opportunity for a significant reduction of the prediction errors, especially if considering an aggregated predictor that takes advantage of the uncorrelatedness of the errors committed by the new predictors and the Terna one.

5.6 Discussion

We have shown that accurate one-day ahead predictions of the Italian electric load demand can be achieved on normal days, by a short-term predictor that does not model yearly seasonality and does not use exogenous information such as the one-day ahead prediction of the temperature. The considered prediction problem consists of predicting tomorrow's quarter-hourly demand profile based on the knowledge of today's profile until midnight. In particular we focused on the development of effective algorithms capable of exploiting the highly correlated nature of the signal. The first steps are a logarithmic transformation to achieve a more stable and symmetric signal and a 7-day differentiation that removes the weekly periodicity. The key idea behind the proposed forecaster is a multipredictor strategy, i.e. developing 96 linear predictors, each of which provides the prediction of the target signal during one of tomorrow's quarter-hours. In other words, each prediction is a linear combination of today's 96 samples. The full model, characterized by $96 \times 96 = 9216$ parameters, is obviously overparametrized so that different regularization approaches were employed to reduce the degrees of freedom without penalizing

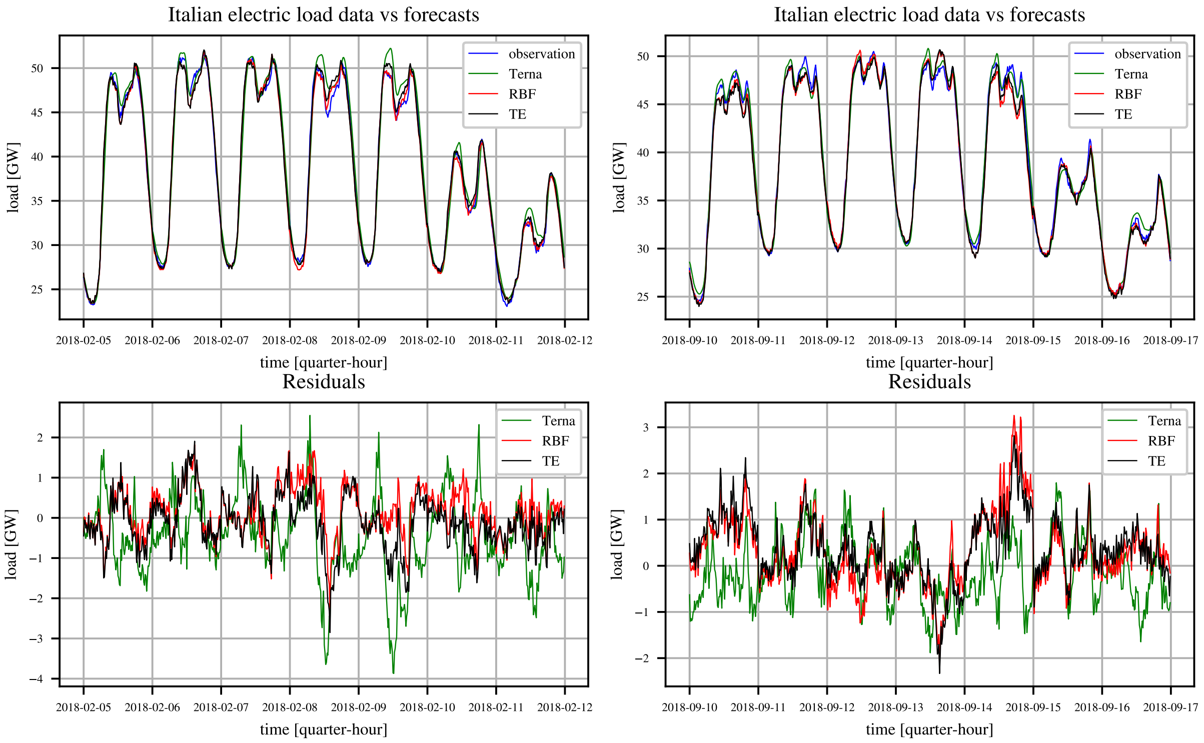


FIGURE 5.12: RBF, TE and Terna prediction (top) and residual (bottom) over two sample weeks on 2018.

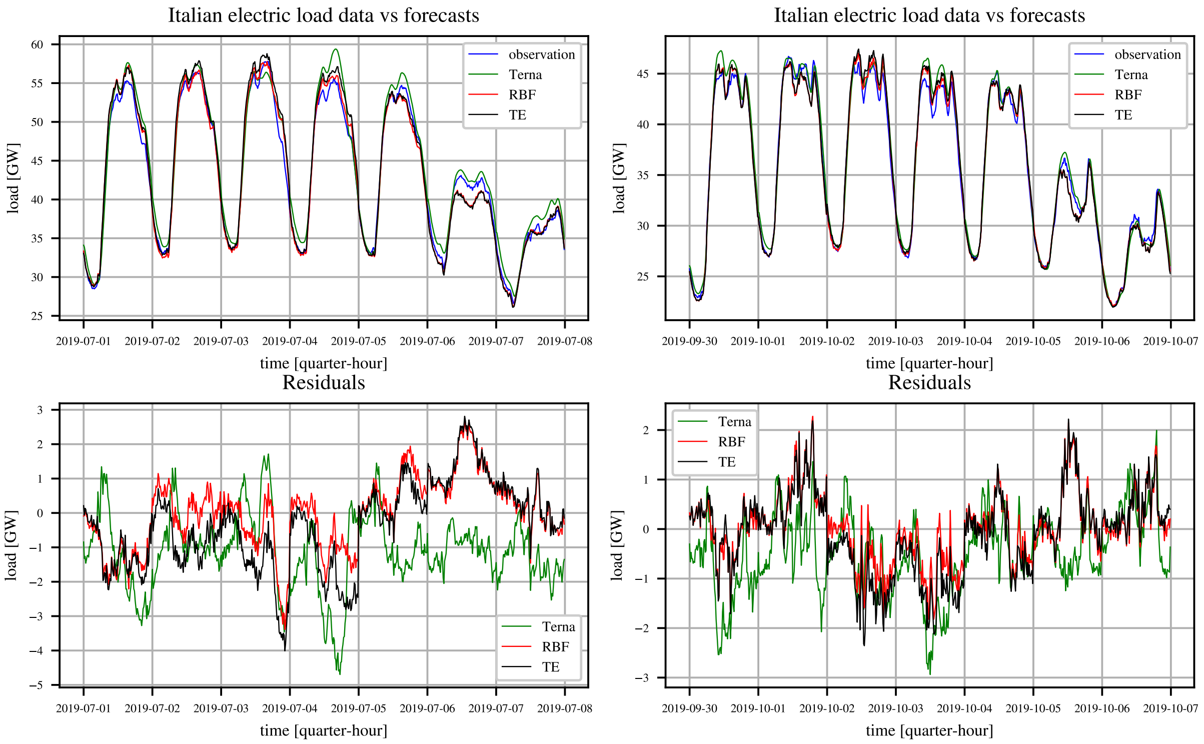


FIGURE 5.13: RBF, TE and Terna prediction (top) and residual (bottom) over two sample weeks on 2019.

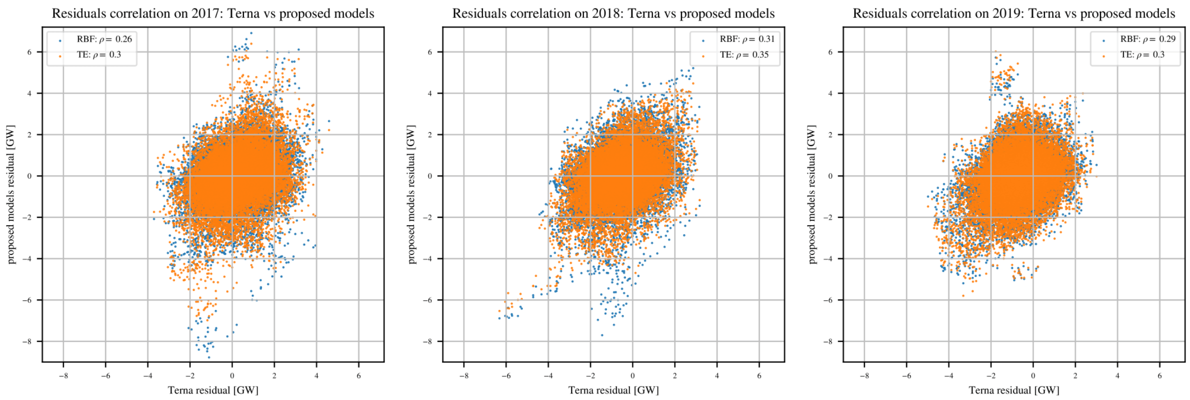


FIGURE 5.14: Scatter plots between the Terna residual and RBF (blue dots) and TE (orange dots) model residuals on the three test scenarios. The Pearson correlation coefficients indicate a weak correlation between the proposed models errors and the Terna forecast error.

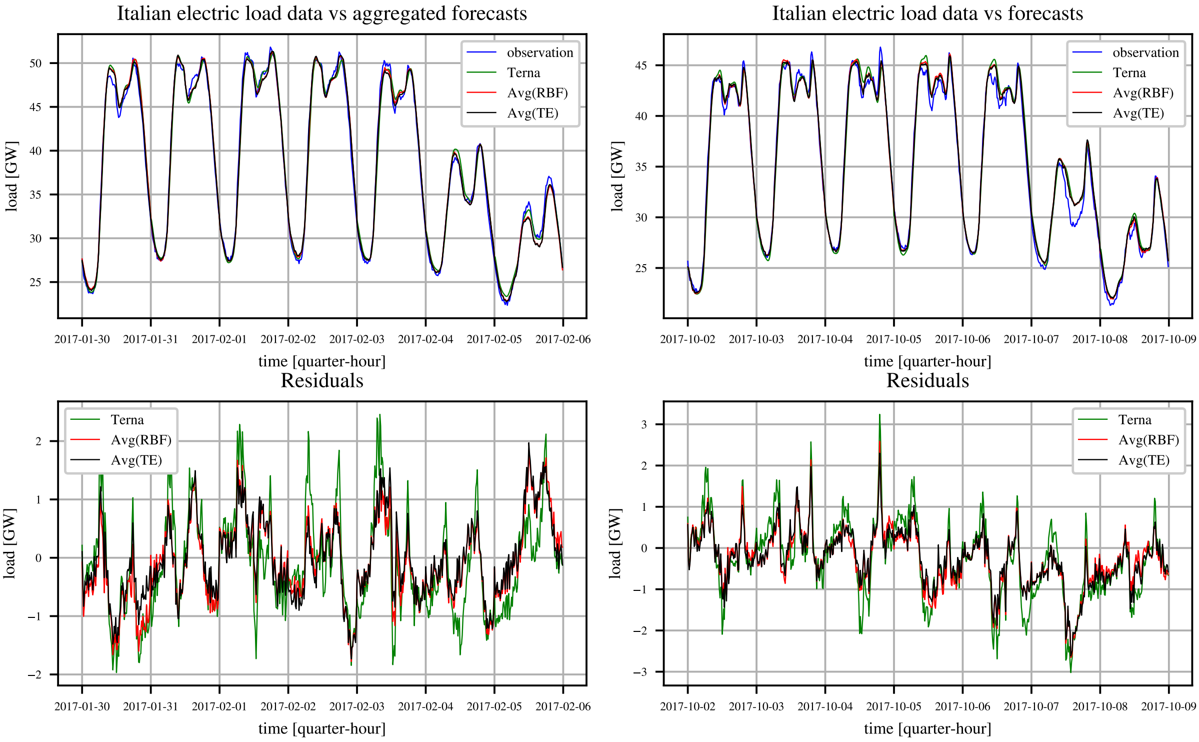


FIGURE 5.15: Avg(RBF), Avg(TE) and Terna prediction (top) and residual (bottom) over two sample weeks on 2017.

Table 5.2: Covariances, biases and Mean Squared Errors of RBF, TE and Terna forecasters, and the corresponding aggregated predictors.

	2017	**2018**	**2019**
$Cov\left[e_{RBF}e_{Terna}\right]$	0.29	0.33	0.29
$Cov\left[e_{TE}e_{Terna}\right]$	0.36	0.36	0.31
$E\left[e_{Terna}\right]$	0.16	-0.52	-0.65
$E\left[e_{RBF}\right]$	-0.03	0.02	-0.06
$E\left[e_{TE}\right]$	-0.04	-0.01	-0.09
MSE_{Terna}	1.13	1.29	1.43
MSE_{RBF}	1.16	1.12	0.98
MSE_{TE}	1.28	1.02	1.09
$M\hat{S}E_{Avg(RBF)}$	0.72	0.77	0.75
$MSE_{Avg(RBF)}$	0.71	0.77	0.77
$M\hat{S}E_{Avg(TE)}$	0.78	0.76	0.79
$MSE_{Avg(TE)}$	0.78	0.76	0.81

TABLE 5.3: Forecast performances on 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\mathbf{daily}}$[%]	**RMSE**$_{\mathbf{daily}}$[**GW**]	**MAE**$_{\mathbf{daily}}$[**GW**]	**dof**
Terna	2.16	**1.06**	0.83	**1.22**	**0.62**	**0.47**	Unknown
LS	2.63 (22%)	1.41 (33%)	1.01 (22%)	1.86 (52%)	0.95 (53%)	0.69 (47%)	9216
TA	1.97 (−9%)	1.07 (1%)	0.75 (−10%)	1.42 (16%)	0.73 (18%)	0.53 (13%)	1686.12
TS	1.97 (−9%)	1.07 (1%)	0.76 (−8%)	1.41 (16%)	0.73 (18%)	0.53 (13%)	138.38
RBF	**1.97 (−9%)**	1.08 (2%)	**0.76 (−8%)**	1.43 (17%)	0.73 (18%)	0.54 (15%)	132.88
TE	2.06 (−5%)	1.13 (7%)	0.79 (−5%)	1.57 (29%)	0.79 (27%)	0.59 (26%)	97.06
OnE	2.64 (22%)	1.4 (32%)	0.99 (19%)	2.04 (67%)	1.12 (81%)	0.77 (64%)	11.12

TABLE 5.4: Forecast performances on 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\mathbf{daily}}$[%]	**RMSE**$_{\mathbf{daily}}$[**GW**]	**MAE**$_{\mathbf{daily}}$[**GW**]	**dof**
Terna	2.41	1.14	0.9	1.65	0.76	0.62	Unknown
LS	2.74 (14%)	1.45 (27%)	1.07 (19%)	2.0 (21%)	0.99 (30%)	0.76 (23%)	9216
TA	1.93 (−20%)	1.07 (−6%)	0.75 (−17%)	1.48 (−10%)	0.74 (−3%)	0.55 (−11%)	1569.78
TS	1.93 (−20%)	1.07 (−6%)	0.75 (−17%)	1.49 (−10%)	0.74 (−3%)	0.55 (−11%)	406.03
RBF	1.92 (−20%)	1.06 (−7%)	0.74 (−18%)	1.46 (−12%)	0.73 (−4%)	0.55 (−11%)	80.49
TE	**1.87 (−22%)**	**1.01 (−11%)**	**0.72 (−20%)**	**1.4 (−15%)**	**0.68 (−11%)**	**0.52 (−16%)**	114.5
OnE	2.32 (−4%)	1.22 (7%)	0.89 (−1%)	1.77 (7%)	0.89 (17%)	0.67 (8%)	4.15

TABLE 5.5: Forecast performances on 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	**MAPE$_{\text{daily}}$**[%]	**RMSE$_{\text{daily}}$**[GW]	**MAE$_{\text{daily}}$**[GW]	**dof**
Terna	2.53	1.2	0.94	1.89	0.85	0.71	Unknown
LS	2.51 (−1%)	1.33 (11%)	0.98 (4%)	1.98 (5%)	0.92 (8%)	0.74 (4%)	9216
TA	1.9 (−25%)	1.02 (−15%)	0.73 (−22%)	1.42 (−25%)	0.72 (−15%)	0.54 (−24%)	477.36
TS	1.84 (−27%)	0.99 (−18%)	0.71 (−24%)	1.37 (−28%)	0.68 (−20%)	0.51 (−28%)	179.04
RBF	**1.84** (**−27%**)	**0.99** (**−18%**)	**0.71** (**−24%**)	**1.36** (**−28%**)	**0.68** (**−20%**)	**0.51** (**−28%**)	78.42
TE	1.91 (−25%)	1.04 (−13%)	0.74 (−21%)	1.48 (−22%)	0.73; (−14%)	0.56 (−21%)	106.37
OnE	2.41 (−5%)	1.25 (4%)	0.92 (−2%)	1.88 (−1%)	0.97 (14%)	0.72 (1%)	3.71

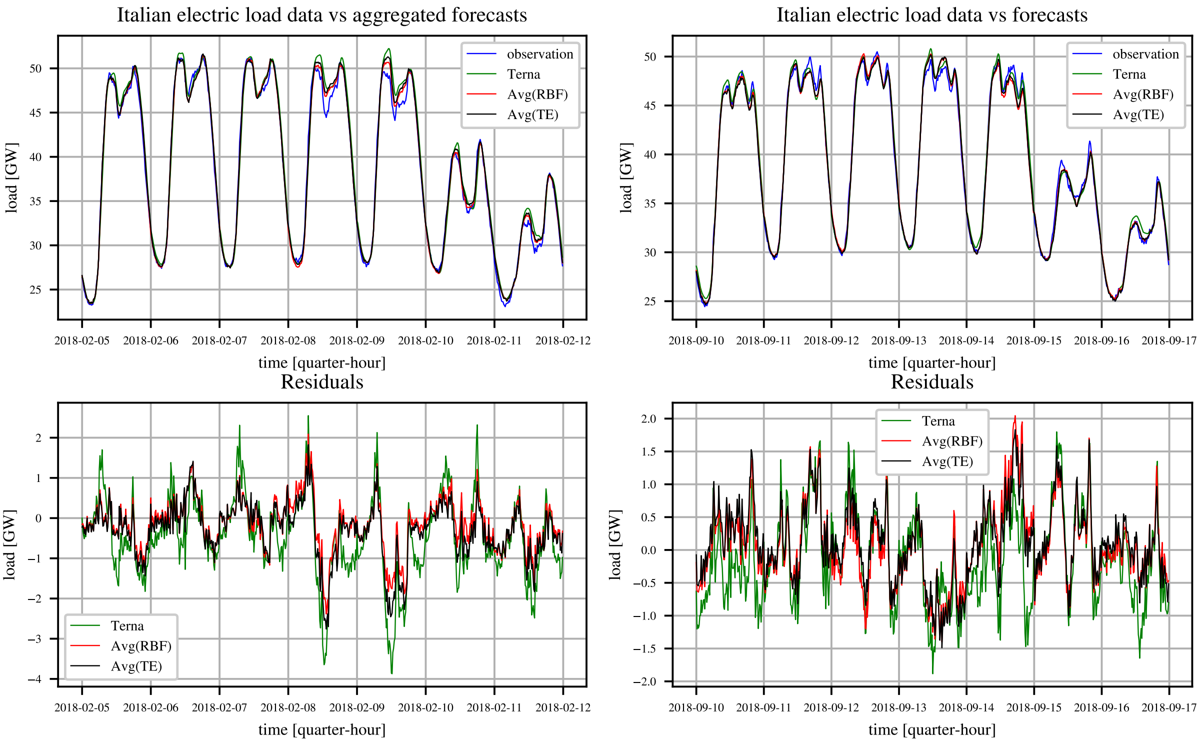


FIGURE 5.16: Avg(RBF), Avg(TE) and Terna prediction (top) and residual (bottom) over two sample weeks on 2018.

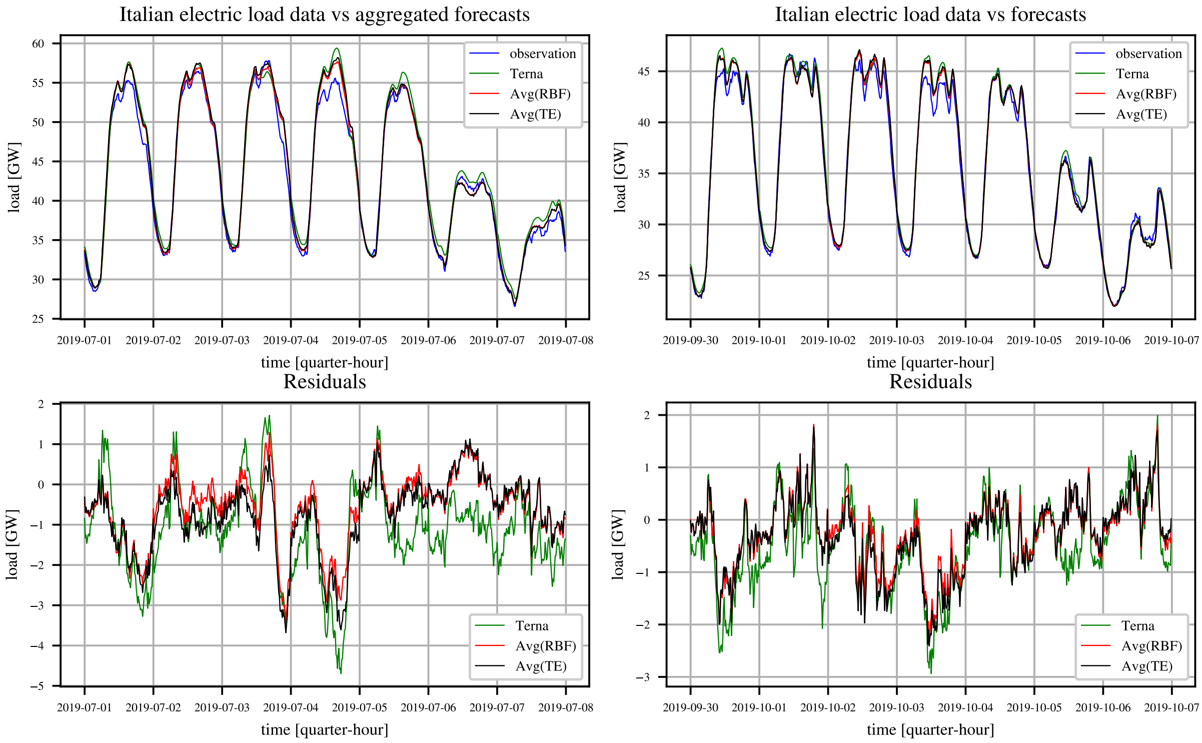


FIGURE 5.17: Avg(RBF), Avg(TE) and Terna prediction (top) and residual (bottom) over two sample weeks on 2019.

TABLE 5.6: Aggregated forecast performances on 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE}_{\mathbf{daily}}$[%]	$\mathbf{RMSE}_{\mathbf{daily}}$[GW]	$\mathbf{MAE}_{\mathbf{daily}}$[GW]
Terna	2.16	1.06	0.83	1.22	0.62	0.47
Avg(LS)	1.87 (−13%)	0.96 (−9%)	0.72 (−13%)	1.19 (−2%)	0.6 (−3%)	0.45 (−4%)
Avg(TA)	1.66 (−23%)	0.85 (−20%)	0.63 (−24%)	1.06 (−13%)	0.53 (−15%)	0.4 (−15%)
Avg(TS)	1.66 (−23%)	0.85 (−20%)	0.63 (−24%)	1.04 (−15%)	0.52 (−16%)	0.39 (−17%)
Avg(RBF)	**1.66 (−23%)**	**0.85 (−20%)**	**0.63 (−24%)**	**1.04 (−15%)**	**0.52 (−16%)**	**0.39 (−17%)**
Avg(TE)	1.72 (−20%)	0.88 (−17%)	0.66 (−20%)	1.13 (−7%)	0.57 (−8%)	0.42 (−11%)
Avg(OnE)	2.04 (−6%)	1.03 (−3%)	0.77 (−7%)	1.44 (18%)	0.76 (23%)	0.55 (17%)

Table 5.7: Aggregated forecast performances on 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[GW]	$\mathbf{MAE_{daily}}$[GW]
Terna	2.41	1.14	0.9	1.65	0.76	0.62
Avg(LS)	1.94 (−20%)	1.01 (−11%)	0.74 (−18%)	1.32 (−20%)	0.66 (−13%)	0.49 (−21%)
Avg(TA)	1.71 (−29%)	0.88 (−23%)	0.65 (−28%)	1.16 (−30%)	0.58 (−24%)	0.43 (−31%)
Avg(TS)	**1.7** (**−29**%)	0.87 (−24%)	0.65 (−28%)	**1.15** (**−30**%)	0.57 (−25%)	**0.43** (**−31**%)
Avg(RBF)	1.71 (−29%)	0.88 (−23%)	0.65 (−28%)	1.16 (−30%)	0.58 (−24%)	0.43 (−31%)
Avg(TE)	1.72 (−29%)	**0.87** (**−24**%)	**0.65** (**−28**%)	1.18 (−28%)	**0.57** (**−25**%)	0.44 (−29%)
Avg(OnE)	1.9 (−21%)	0.96 (−16%)	0.72 (−20%)	1.34 (−19%)	0.67 (−12%)	0.5 (−19%)

Table 5.8: Aggregated forecast performances on 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[GW]	$\mathbf{MAE_{daily}}$[GW]
Terna	2.53	1.2	0.94	1.89	0.85	0.71
Avg(LS)	1.95 (−23%)	0.99 (−18%)	0.74 (−21%)	1.45 (−23%)	0.69 (−19%)	0.54 (−24%)
Avg(TA)	1.76 (−30%)	0.89 (−26%)	0.67 (−29%)	1.28 (−32%)	0.63 (−26%)	0.48 (−32%)
Avg(TS)	1.73 (−32%)	0.88 (−27%)	0.65 (−31%)	1.23 (−35%)	0.6 (−29%)	0.46 (−35%)
Avg(RBF)	**1.73** (**−32**%)	**0.88** (**−27**%)	**0.65** (**−31**%)	**1.23** (**−35**%)	**0.6** (**−29**%)	**0.46** (**−35**%)
Avg(TE)	1.78 (−30%)	0.9 (−25%)	0.67 (−29%)	1.3 (−31%)	0.64 (−25%)	0.49 (−31%)
Avg(OnE)	1.98 (−22%)	1.0 (−17%)	0.75 (−20%)	1.51 (−20%)	0.74 (−13%)	0.57 (−20%)

the predictive capabilities. The main observation is that the 9216 parameters can be represented as a surface that, in view of the regularity of the load signal, should exhibit some smoothness properties.

The test results over 2017-2019 have shown that, through a wise application of regularization techniques it is possible to obtain competitive predictors whose MAPEs improve on that of Terna. Moreover, the residuals of the proposed predictors are weakly correlated with Terna's, suggesting that aggregated forecasters could further improve the final results.

As a matter of fact, averaging the Terna predictions with the proposed forecasts allows to reach an improvement up to the 30% with respect to the Terna benchmark forecast in all performance indexes, both in a quarter-hourly and daily framework.

6

Short-term forecasting of the load demand during the Easter Week

One of the challenges of electric load forecasting is given by ‘so-called’ intervention events. According to Box et al. “Time series are often affected by special events or conditions such as policy changes, strikes, advertising promotions, environmental regulations, and similar events, which we will refer to as intervention events” [94]. In the context of load forecasting, holidays and phenomena out of the ordinary such as blackouts and lockdowns can be treated as instances of intervention events. In particular, the different behaviours observed during these periods compared to ordinary ones motivates the classification of days as normal and special ones.

During a special day the typical seasonal and weekly patterns characterizing the load demand on normal days is blurred: the load profile tends to change its shape, typically assuming lower values because of the closure of activities and industries, whose electrical demand usually covers a significant fraction of the overall demand. Therefore, when addressing the prediction of the load demand on special days, it is not safe to rely on the same forecasting models used on normal days and, rather, it is convenient to build *ad hoc* forecasters.

As reported in Appendix A, Italian Holidays can be divided in: (i) Winter holidays, Christmas, New Year and Epiphany; (ii) Summer holidays, three weeks around August 15; (iii) national holidays, April 25, May 1, June 2, November 1, December 8; (iv) Easter. Among these, Easter represents a particular case, since its date is not fixed but changes every year according to the ‘computus’ [95] and it may fall anytime between March and April. This characteristic makes Easter load demand shape more volatile, since different seasonal phenomena can affect the load depending on the particular date. For instance, in a lot of countries, Daylight Saving Time begins around the end of March and affects the load profile shape by shifting the demand with respect to time [96].

A variety of strategies have been proposed in the literature to deal with the forecast of load demand during the Easter holidays, such as multiple linear regression [97], similar day approach [70], clustering-based methods [98] and artificial neural networks [99]. However, the first approach does not take into account possible nonlinearities while the similar day method, in order to be effective, may require the availability of several years of data as reference. For what concerns the third strategy, the problem of finding an exact way for defining the boundaries that separate the clusters remains open. Finally, artificial neural network algorithms are characterized by long training time and low interpretability. In this chapter, with reference to the Italian case, a novel one-day ahead forecaster based on Gaussian Process Regression with a 'custom' kernel is developed for the prediction of the load demand during the Easter holidays.

6.1 Dataset and preprocessing

The dataset considered in this chapter consists of:

- the 25-year series (from 1990 to 2014) of quarter-hourly Italian load demand coming from a historical database (the same considered in Section 4.1).
- the 5-year long time series of quarter-hourly Italian electric load demands (from 2015 to 2019), downloaded from the Transparency Report Platform of Terna available at [87] (the same considered in Section 5.1), whose time plot is displayed again in Fig. 6.1 for ease of reference.
- the 3-year long time series of quarter-hourly forecasts elaborated by Terna (from 2017 to 2019), available as well in [87] and already considered in Section 5.1.

For the same considerations made in Section 5.1, a logarithmic transformation is applied to the data as a first preprocessing operation.

Before extracting the load data associated to Easter days, it is necessary to remove the trend component so that the Easter Week profiles are not affected by differences in the mean which are mainly due to economical and social factors characterizing the different years. For this purpose, the trend component is modelled and then removed from the full log-transformed profile.

In particular, let $L(d, q)$ denote the country load demand at the q-th quarter-hour of the day d, where $1 \leq q \leq 96$ and d is an integer serial representing the whole number of days from a fixed, preset date (e.g. January 0, 0000) in the proleptic ISO calendar. Then

$$S(d, q) := \ln\left(L(d, q)\right) = S_{detr}(d, q) + T(d) + \epsilon$$

where $S_{detr}(d, q)$ is the log-transformed load demand without trend component, $T(d)$ is the trend component and ϵ is a noise term.

The trend component of the 25-year log-transformed series was estimated using a lowpass filter with bandwidth 1/730 day^{-1} (see 4.1).

A linear function is adopted to estimate the trend component in the more recent 5-year dataset. This choice is driven by the fact that the trend term includes, by definition, very low frequency components of the load demand and does not exhibit a 'more-than-linear' behaviour in the considered 5-year timespan.

The linear function estimate of $T(d)$ is given by:

$$\hat{T}(d) = \beta + \alpha d$$

where the parameters β and α are estimated by means of the Least Squares algorithm. The resulting trend $\hat{T}(d)$, shown in red in Fig. 6.2 (top), is almost constant, as confirmed by the estimate of parameters, $\beta = 3.56$ (intercept) and $\alpha = 9.27 \times 10^{-8}$ (slope) (estimate $\pm 2.23 \times 10^{-8}$ SE). The associated detrended log-load time series

$$S_{detr}(d, q) = L(d, q) - \hat{T}(d)$$

is shown in the bottom panel of Fig. 6.2

Finally, let $\mathcal{E} \subset \mathcal{D}_s$ be the set of all the Easter Week days within the range 2015-2019. Then:

$$Y_{\mathcal{E}}(d, q) = \begin{cases} \text{missing}, & \text{if } (d \notin \mathcal{E}) \\ S_{detr}(d, q), & \text{otherwise} \end{cases}$$

where $\mathcal{D}_s$ and the dates of the days belonging to $\mathcal{E}$ are reported in Appendix A. Notice that, in this work, the Easter Week goes from the Thursday preceeding Easter Day to the Easter Monday, for a total of 5 days for each Easter Week.

In Fig. 6.3, the superimposition of all the quarter-hourly Easter Week detrended log-load demand profiles is shown. The plot highlights the high repeatability of the normalized Easter time series over the years, even though each profile has its own features.

6.2 Problem statement

Let define $t = 1, \dots, 480$ as the quarter-hour within an Easter-Week, e.g. $t = 1$ is the 1^{st} quarter-hour of Thursday and $t = 480$ is the 96^{th} quarter-hour of Monday.

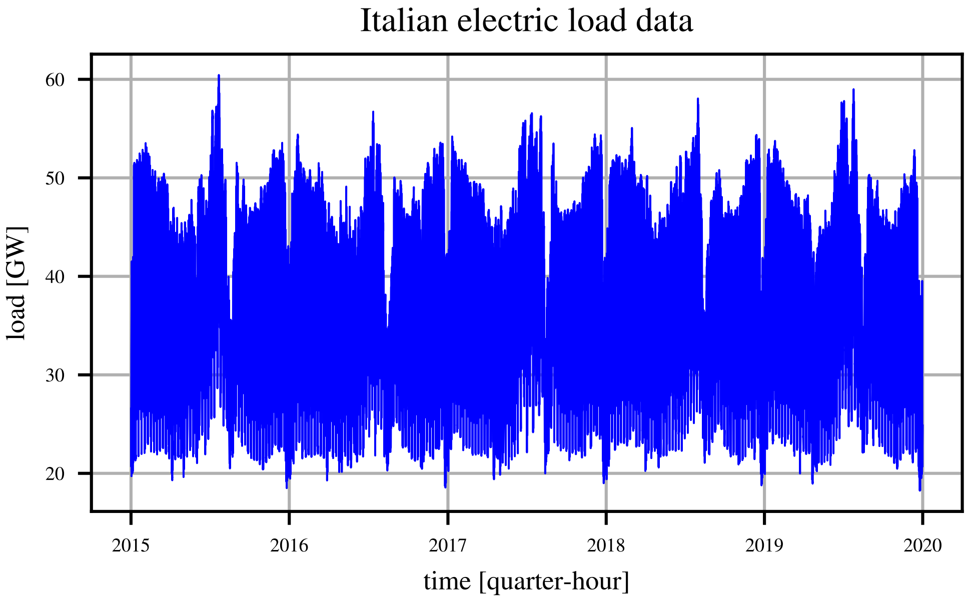


FIGURE 6.1: Italian quarter-hourly electric load demand from 2015 to 2019.

The goal of this chapter is to build a predictor $\hat{L}_{\mathcal{E}}(t)$ for the Easter Week load demand $L_{\mathcal{E}}(t)$. Notice that

$$\hat{L}_{\mathcal{E}}(t) = \exp\left(\hat{Y}_{\mathcal{E}}(t) + \hat{T}_{\mathcal{E}}(t)\right)$$

where $\hat{T}_{\mathcal{E}}(t)$ is the prediction of the trend and $\hat{Y}_{\mathcal{E}}(t)$ is the prediction of the detrended log-load demand over the Easter Week.

Leveraging the high repeatability of $Y_{\mathcal{E}}(t)$, the key idea is to model the 'average' Easter profile and then develop a recursive short-term strategy to 'track' the shift of the load demand of the current Easter with respect to the average profile. On the basis of this consideration, the detrended log-load demand during the Easter Week $Y_{\mathcal{E}}(t)$ can be written as

$$Y_{\mathcal{E}}(t) = \bar{Y}_{\mathcal{E}}(t) + \tilde{Y}_{\mathcal{E}}(t) + v$$

where $\bar{Y}_{\mathcal{E}}(t)$ is the average profile, common to all years, $\tilde{Y}_{\mathcal{E}}(t)$ is a shift term specific of the current year, and v is a noise term.

Let

$$\hat{Y}(t|\tau) = E[Y(t)|Y(1), \dots, Y(\tau)]$$

denote the mean squared estimate of Y, given $Y(1), \dots, Y(\tau)$. Assuming that an external estimate, e.g. based on historical data, $\hat{\bar{Y}}_{\mathcal{E}}(t)$ is available, then $\hat{Y}_{\mathcal{E}}(t|\tau)$ can be estimated as

$$\hat{Y}_{\mathcal{E}}(t|\tau) = \hat{\bar{Y}}_{\mathcal{E}}(t) + \hat{\tilde{Y}}_{\mathcal{E}}(t|\tau)$$

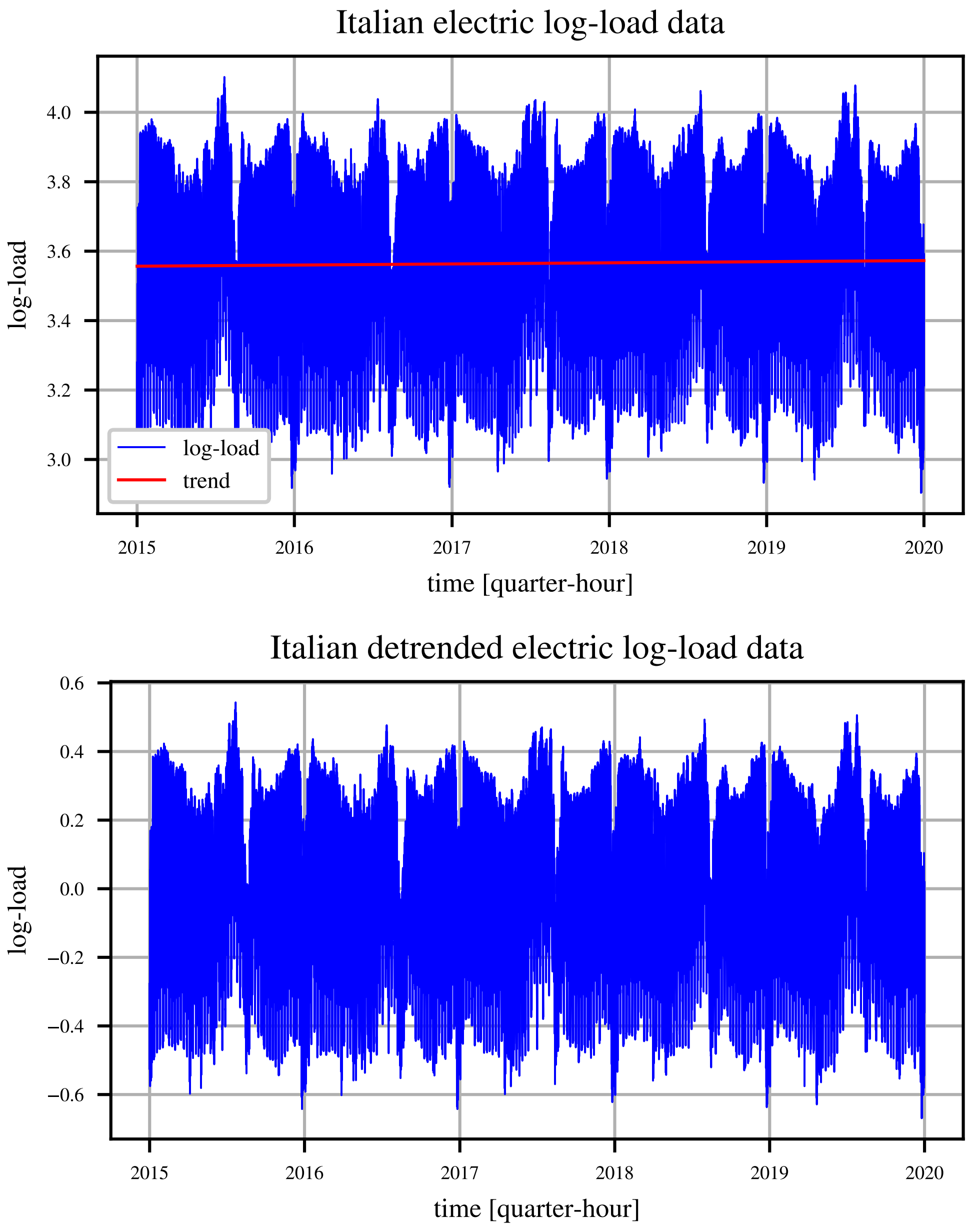


FIGURE 6.2: Preprocessed Italian quarter-hourly electric load demand: log-transformed time series S in blue and estimated linear trend $\hat{T}$ in red (top) and detrended log-transformed time series S_{detr} (bottom).

Note also that, given $\hat{\hat{Y}}_{\mathcal{E}}(t)$, we have

$$\tilde{Y}_{\mathcal{E}}(t) = Y_{\mathcal{E}}(t) - \hat{\hat{Y}}_{\mathcal{E}}(t).$$

Then, the Easter forecasting problem amounts to solving the following one-day ahead five prediction problems. Find

$$\hat{\hat{Y}}_{\mathcal{E}}(t|k96), \quad k96 + 1 \leq t \leq (k+1)96$$

for the following values of k:

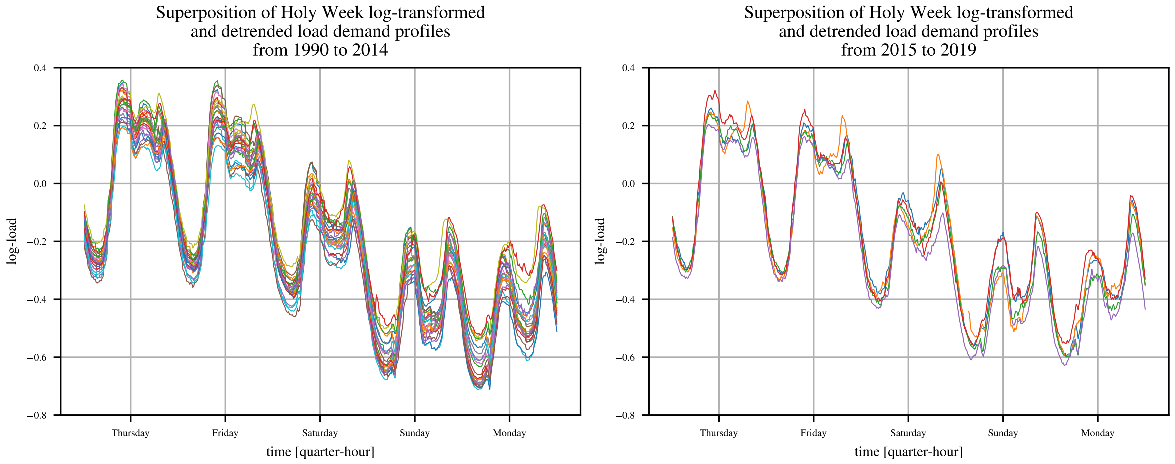


FIGURE 6.3: Detrended log-load demands of the Easter Weeks from 1990 to 2014 (left) and from 2015 to 2019 (right).

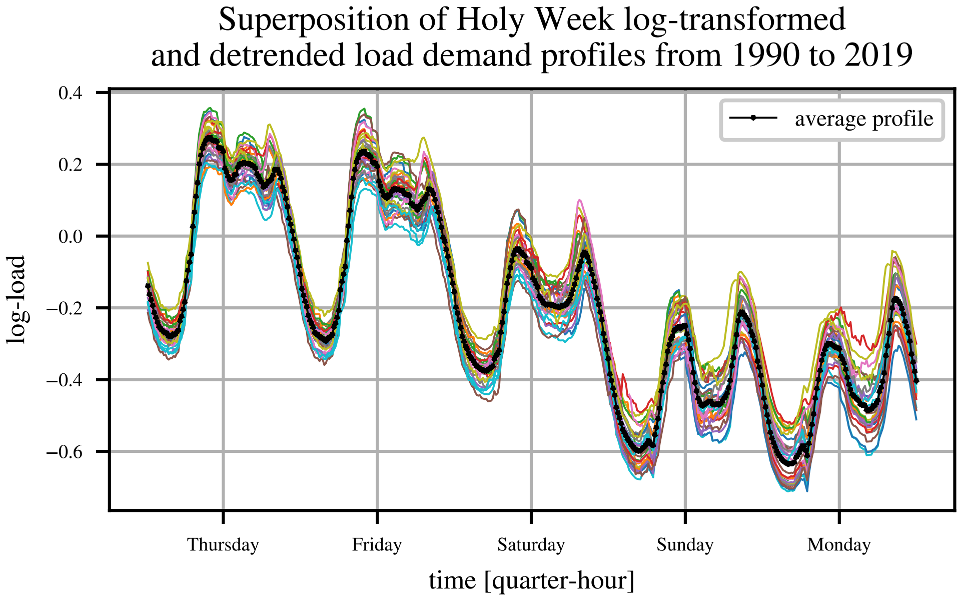


FIGURE 6.4: Detrended log-load demands of the Easter Weeks from 1990 to 2019 and corresponding average profile (black).

1. $k = 0$ (Thursday)

2. $k = 1$ (Friday)

3. $k = 2$ (Saturday)

4. $k = 3$ (Sunday)

5. $k = 4$ (Monday)

It goes without saying that, for $k = 0$, one has the trivial solution $\hat{\hat{Y}}_{\mathcal{E}}(t|0) = 0$, $1 \leq t \leq 96$. It is immediate to see that, from Thursday to Sunday, the objective is predicting tomorrow's 96 load shifts, based on the data up to midnight of the current day.

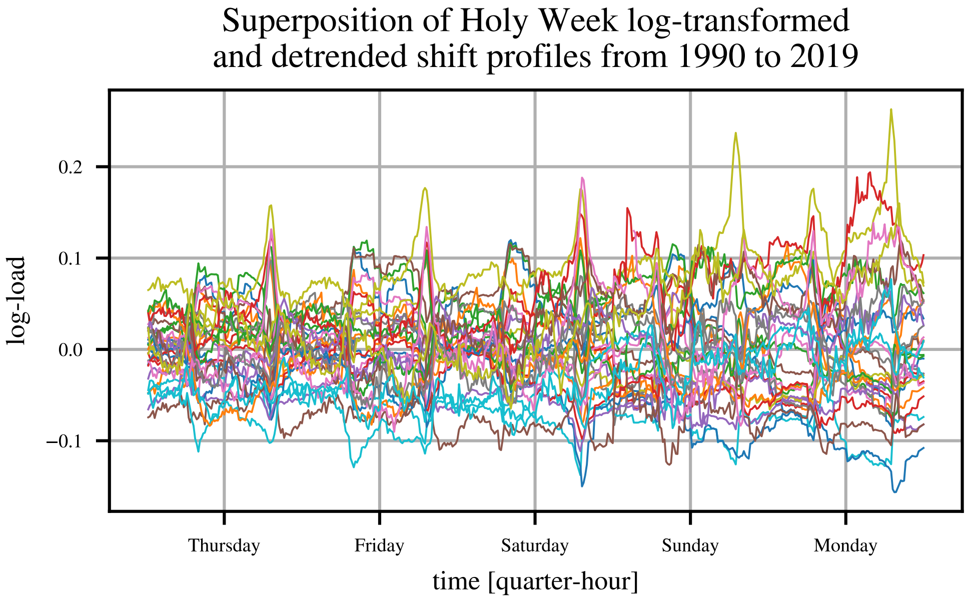


FIGURE 6.5: Easter Week detrended log-load demands shifts from 1990 to 2019.

6.3 Model estimation

The most straightforward approach for modelling $\bar{Y}_{\mathcal{E}}(t)$ is to compute the average of the available Easter Week profiles for each quarter-hour of each day, that is

$$\hat{\bar{Y}}_{\mathcal{E}}(t) = \frac{1}{n_y} \sum_i Y_{\mathcal{E}}^i(t)$$

where n_y is the number of years considered and $Y_{\mathcal{E}}^i(t)$ is the detrended log-transformed observations of the Easter Week of the $i - th$ year. The average profile estimated over the years 1990-2019 is displayed in black in Fig. 6.4 together with the Easter Week observations of the same years.

The 'shift' term $\tilde{Y}_{\mathcal{E}}(t)$ is obtained by subtracting the average profile $\hat{\bar{Y}}_{\mathcal{E}}(t)$ from $Y_{\mathcal{E}}(t)$.

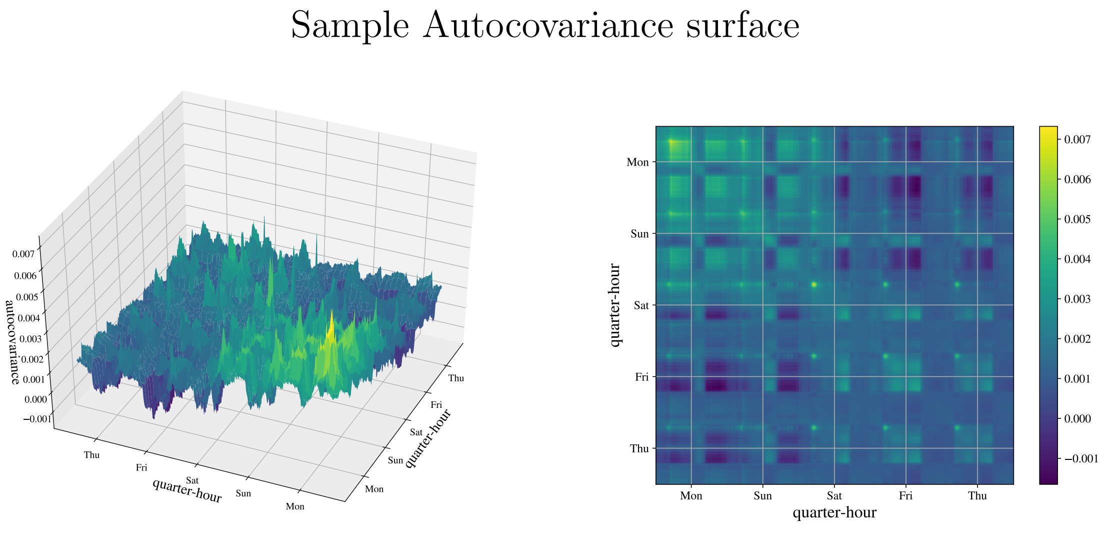
In Fig. 6.5 the shift profiles $\tilde{Y}_{\mathcal{E}}^i(t)$, obtained by subtracting the average profile $\hat{\bar{Y}}_{\mathcal{E}}(t)$ from each $Y_{\mathcal{E}}^i(t)$, $i = 2015, \ldots, 2019$, are displayed.

In order to derive a short-term predictive model it is assumed that the shift time series of Fig. 6.5 are realizations of a zero Gaussian process plus noise, that is:

$$\tilde{Y}_{\mathcal{E}}(t) = f(t) + \eta$$

$$f(t) \sim \mathcal{N}\left(0, k(t, t')\right)$$

where η is an independent identically distributed Gaussian noise with variance σ^2 and $k(t, t') = E[f(t)f(t')]$ is the autocovariance function, also called 'kernel', that completely specifies the



(a) 3D view

(b) Top view

FIGURE 6.6: Sample Autocovariance surface computed on the Easter shift profiles over the years 1990-2014.

Gaussian process and encodes all the prior informations. The choice of the kernel is crucial in order to optimize the results of regression or classification problems and the literature offers a great variety of possible choices, see [54] for an exhaustive overview.

In this work, instead, a different approach is pursued. Rather than picking up a kernel from the literature, the availability of extensive historical data is exploited in order to directly estimate the autocovariance $k(t, t')$ of the shift profiles.

This implies that $k(t, t') = \hat{r}(t, t')$, where the sample autocorrelation (that coincides with the autocovariance because the stochastic process has zero mean) is given by

$$\hat{r}(t, t') = \frac{1}{y_f - y_0 + 1} \sum_{i=y_0}^{y_f} \tilde{Y}^i_{\mathcal{E}}(t)\tilde{Y}^i_{\mathcal{E}}(t'), \quad 1 \leq t \leq 480, \quad 1 \leq t' \leq 480 \tag{6.1}$$

where y_0 and y_f denote the initial and final year used to estimate the autocorrelation.

Once $\hat{r}(t, t')$ has been computed, the so-called kernel matrix

$$K = [K]_{ij} \in \mathbb{R}^{480 \times 480}, K_{ij} = \hat{r}(t_i, t_j)$$

can be displayed as a bidimensional surface, see Fig. 6.6, where the sample autocorrelation was computed with $y_0 = 1990$ and $y_f = 2014$.

Before proceeding with the solution of the Easter forecasting problem, it is convenient to recall the representer theorem for Gaussian processes [54].

Theorem. [54] Let $y(t) = f(t) + \eta(t)$, where $f(\cdot)$ is a discrete-time zero mean Gaussian Process with $E\left[f(t)f(t')\right] = k(t,t')$ and $\eta(t) \sim WN(\sigma^2)$ a white noise independent of $f(\cdot)$. Consider the problem of estimating $f(\tau)$, given the knowledge of n observations $\{y(\tilde{t}_i)\}$ sampled at times $\{\tilde{t}_i\}$, $i = 1, \ldots, n$. Then the optimal (mean square) estimate is given by the conditional expectation

$$\hat{f}(\tau) = E\left[f(\tau)|y(\tilde{t}_1), \ldots, y(\tilde{t}_n)\right] = \sum_i \alpha_i k(\tau, \tilde{t}_i)$$

where

$$\begin{aligned}
&\alpha = (\tilde{K} + \sigma^2 I)^{-1} y \\
&\tilde{K} = [\tilde{K}]_{ij} \in \mathbb{R}^{n \times n}, \tilde{K}_{ij} = k(\tilde{t}_i, \tilde{t}_j) \\
&y = \left[\begin{array}{ccc} y(\tilde{t}_1) & \ldots & y(\tilde{t}_n) \end{array}\right]^T
\end{aligned}$$

Proof. By a well known formula for the conditional expectation of jointly normal random variables

$$\begin{aligned}
&E\left[f(\tau)|y(\tilde{t}_1), \ldots, y(\tilde{t}_n)\right] = \\
=&\left[\begin{array}{ccc} E\left[f(\tau), y(\tilde{t}_1)\right] & \ldots & E\left[f(\tau), y(\tilde{t}_n)\right] \end{array}\right] Var[y]^{-1} y = \\
=&\left[\begin{array}{c} k(\tau, \tilde{t}_1) \ldots k(\tau, \tilde{t}_n) \end{array}\right] (\tilde{K} + \sigma^2 I)^{-1} y \quad \square
\end{aligned}$$

In view of the representer theorem, the prediction $\hat{f}(\tau_k) = \hat{\tilde{Y}}_{\mathcal{E}}(\tau_k|k96)$, $\tau_k = k96+1, \ldots, (k+1)96$, $k = 1, 2, 3, 4$, is given by the following formula:

$$\hat{f}(\tau_k) = \sum_{i=1}^{k96} \alpha_i^k k(\tau_k, i) \tag{6.2}$$

with

$$\begin{aligned}
&\alpha^k = (\Delta^k + \sigma^2 I)^{-1} y^k \\
&\Delta^k = [\Delta^k]_{ij} \in \mathbb{R}^{k96 \times k96}, \Delta^k_{ij} = k(i,j), \quad 1 \le i, j \le k96 \\
&y^k = \left[\begin{array}{ccc} y(1) & \ldots & y(k96) \end{array}\right]^T
\end{aligned}$$

It is worth noting that Δ^k are submatrices of the full kernel matrix K. In the procedure, the only hyperparameter that needs to be calibrated is the standard deviation σ, see Section 6.4.

Of course, the formula (6.2) has to be computed for $k = 1, \ldots, 4$, namely for each target Easter day from Friday to Monday.

The last open question is how to forecast the Thursday quarter-hourly detrended log-transformed profile $Y_{\pounds}(\tau_0)$.

Four possible strategies are:

- Average Profile (AP)
- Similar Day (SD)
- Last Year (LY)
- Normal days short-term prediction

The AP strategy consists of predicting the detrended log-load of Thursday as the average of the detrended log-load of Thursday of the Easter Weeks of the previous years.

$$\hat{Y}_{\pounds}(\tau_0) = \bar{Y}_{\pounds}(\tau_0)$$

The SD strategy consists of predicting the detrended log-load of Thursday as the detrended log-load of Thursday of a similar Easter Week, that satisfies two conditions:

- Easter falls in a day of year that is closest to the one of this year's Easter.
- The two Easters are not separated by the Daylight Saving Time switch.

$$\hat{Y}_{\pounds}(\tau_0) = Y_{\pounds}^{SD}(\tau_0)$$

The LY strategy consists of predicting the detrended log-load of the Easter Thursday as the detrended log-load of the Easter Thursday of the previous year.

$$\hat{Y}_{\pounds}(\tau_0) = Y_{\pounds}^{LY}(\tau_0)$$

While these three methods rely on long-term forecast strategies, the last strategy is based on short-term forecasting. In particular, the RBF forecasting method described in 5.3.3 is used to predict the load demand of the Easter Thursday.

$$\hat{L}_{\pounds}(\tau_0) = L_{\pounds}^{RBF}(\tau_0)$$

For the first three methods, the prediction of the trend component $\hat{T}_{\mathbb{E}}(t)$, $t = 1, \dots, 480$, for the target Easter Week profile is taken as a constant value equal to the trend estimated up to the day before the Easter Week starts.

For the same considerations explained in details in 5.3.6, aggregation methods offer an appealing way to improve performances of existing methods. Therefore, the aggregation between our Easter forecasts and the Terna ones was computed and assessed. For the considered case, the aggregation procedure coincides with the following one with $m = 2$:

Aggregate prediction: let $\mathcal{M}_i, i = 1, \dots, m$, denote a set of prediction methods and $\hat{L}_{\mathcal{M}_i}(t)$ the corresponding load demand forecasts generated by $\mathcal{M}_i$. Then

$$\hat{L}_{\text{Avg}}(t) = \frac{1}{m} \sum_{i=1}^{m} \hat{L}_{\mathcal{M}_i}(t)$$

is the aggregate forecast obtained by averaging the predictions generated by $\mathcal{M}_i$.

6.4 Experimental validation setup

The predictor performances were evaluated over the Easter Weeks of years 2017, 2018 and 2019 and compared to the ones of the Terna forecaster. For each test year, the Easter Week observations of the previous years were used for the training phase, while the calibration of the hyperparameter σ was performed through cross-validation. In particular, for each scenario, the observations of the year preceding the test one were used for the validation step. For all the three test years, the order of magnitude was $\sigma = 0.1$ (2017: $\sigma = 0.22$, 2018: $\sigma = 0.18$, 2019: $\sigma = 0.32$). The performances of the predictors were evaluated using the Mean Absolute Percentage Error ($MAPE$), the Root Mean Square Error ($RMSE$), and the Mean Absolute

Error (MAE), each one on both quarter-hourly and daily data.

$$MAPE = \frac{100}{480} \sum_{d \in \mathcal{E}_{Te}} \sum_{q=1}^{96} \left| \frac{L(d,q) - \hat{L}(d,q)}{L(d,q)} \right|$$

$$RMSE = \sqrt{\frac{\sum_{d \in \mathcal{E}_{Te}} \sum_{q=1}^{96} \left(L(d,q) - \hat{L}(d,q) \right)^2}{480}}$$

$$MAE = \frac{\sum_{d \in \mathcal{E}_{Te}} \sum_{q=1}^{96} \left| L(d,q) - \hat{L}(d,q) \right|}{480}$$

$$MAPE_{daily} = \frac{100}{5} \sum_{d \in \mathcal{E}_{Te}} \left| \frac{L^{daily}(d) - \hat{L}^{daily}(d)}{L^{daily}(d)} \right|$$

$$RMSE_{daily} = \sqrt{\frac{\sum_{d \in \mathcal{E}_{Te}} \left(L^{daily}(d) - \hat{L}^{daily}(d) \right)^2}{5}}$$

$$MAE_{daily} = \frac{\sum_{d \in \mathcal{E}_{Te}} \left| L^{daily}(d) - \hat{L}^{daily}(d) \right|}{5}$$

where $\mathcal{E}_{Te}$ is the test set of Easter Week days and L^{daily} and $\hat{L}^{daily}$ are respectively the daily averages of electric load observations and of associated forecasts:

$$L^{daily}(d) = \frac{1}{96} \sum_{q=1}^{96} L(d,q), \quad \hat{L}^{daily}(d) = \frac{1}{96} \sum_{q=1}^{96} \hat{L}(d,q)$$

6.5 Performances

The forecasters introduced in Section 6.3 were labelled as follows:

- GP_{AP}: Gaussian Process with Average Profile Thursday forecast.
- GP_{SD}: Gaussian Process with Similar Day Thursday forecast.
- GP_{LY}: Gaussian Process with Last Year Thursday forecast.
- GP_{RBF}: Gaussian Process with Radial Basis Function Thursday forecast.

The results for the three test years are summarized in Tables 6.1, 6.2, 6.3.

In the 2017 test year, all the proposed strategies performed better than Terna, with percentage improvements up to 15%, 24% and 19% respectively for quarter-hourly MAPE, RMSE and

Table 6.1: Year 2017: forecasting performances and percentage variation with respect to Terna (between brackets).

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[**GW**]	$\mathbf{MAE_{daily}}$[**GW**]
Terna	2.09	0.84	0.63	1.53	0.49	0.45
$\mathbf{GP_{AP}}$	1.82 (−13%)	0.68 (−19%)	0.52 (−17%)	**0.56** (−**63**%)	**0.18** (−**63**%)	**0.17** (−**62**%)
$\mathbf{GP_{SD}}$	1.77 (−15%)	0.64 (−24%)	0.51 (−19%)	0.64 (−57%)	0.21 (−56%)	0.19 (−57%)
$\mathbf{GP_{LY}}$	**1.77** (−**15**%)	**0.64** (−**24**%)	**0.51** (−**19**%)	0.64 (−57%)	0.21 (−56%)	0.19 (−57%)
$\mathbf{GP_{RBF}}$	1.81 (−13%)	0.64 (−24%)	0.51 (−19%)	0.74 (−52%)	0.27 (−45%)	0.23 (−49%)

MAE, achieved by the GP_{LY} forecaster. In the daily framework, the GP_{AP} predictor outperformed all the other models achieving improvements up to 62-65% for all indexes. This is due to the fact that the quarter-hourly residual of the GP_{AP} on Thursday exhibits positive values in the peak from 13:00 to 18:00 and negative ones until 20:00, see Fig. 6.7, leading to a compensation of the daily error when upsampling the forecast.

The prediction of the 2018 Easter Week load demand, shown in Fig. 6.8, was characterized by smaller percentage improvements (10% for MAPE, 11% MAE, 3% for RMSE) achieved in the quarter-hourly framework by the proposed model with the Average Profile strategy on Easter Thursday. In the daily case the short-term RBF approach for forecasting the Thursday load makes the difference for MAPE and MAE with respect to the other proposed strategies, even though in the daily RMSE Terna still achieves the best performances.

In the 2019 test year, shown in Fig. 6.9, the GP_{RBF} strategy brings improvement with respect to Terna on all indexes, both quarter-hourly (22%, 9%, 18% respectively for MAPE, RMSE, MAE) and daily (14%, 5%, 16% respectively for MAPE, RMSE, MAE). The superiority of GP_{RBF} is due to the more accurate forecast on Easter Thursday.

Both the scatter plots and the correlation coefficients between the residuals of our methods and Terna's one, see Fig. 6.10, reveal a scarce correlation, which suggests the existence of interesting improvement margins exploitable by the simple aggregation of the two predictors, in particular those produced by the GP_{RBF} model. The performances of the aggregated predictors, namely $Avg(GP_{AP})$, $Avg(GP_{SD})$, $Avg(GP_{LY})$ and $Avg(GP_{RBF})$ are reported in Tables 6.4, 6.5 and 6.6 and can be visually inspected in Fig. 6.11, 6.12 and 6.13.

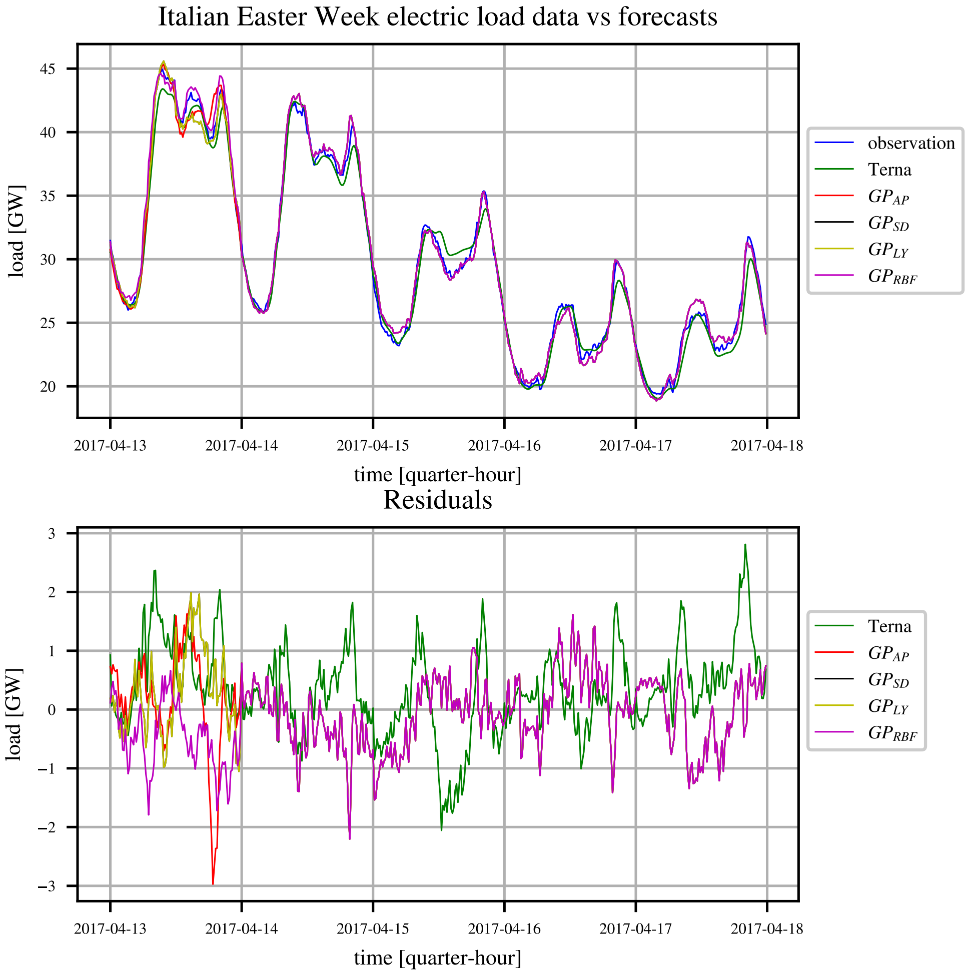


FIGURE 6.7: Easter Week of 2017: data vs forecasts and residuals.

TABLE 6.2: Year 2018: forecasting performances and percentage variation with respect to Terna (between brackets).

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\textbf{daily}}$[%]	**RMSE**$_{\textbf{daily}}$[**GW**]	**MAE**$_{\textbf{daily}}$[**GW**]
Terna	3.2	1.18	0.99	2.06	**0.68**	0.62
GP$_{\textbf{AP}}$	**2.88** (**−10**%)	**1.14** (**−3**%)	**0.88** (**−11**%)	2.21 (7%)	0.78 (15%)	0.64 (3%)
GP$_{\textbf{SD}}$	3.01 (−6%)	1.14 (−3%)	0.92 (−7%)	2.16 (5%)	0.76 (12%)	0.62 (0%)
GP$_{\textbf{LY}}$	2.95 (−8%)	1.17 (−1%)	0.91 (−8%)	2.3 (12%)	0.83 (22%)	0.67 (8%)
GP$_{\textbf{RBF}}$	2.99 (−7%)	1.17 (−1%)	0.9 (−9%)	**2.04** (**−1**%)	0.72 (6%)	**0.57** (**−8**%)

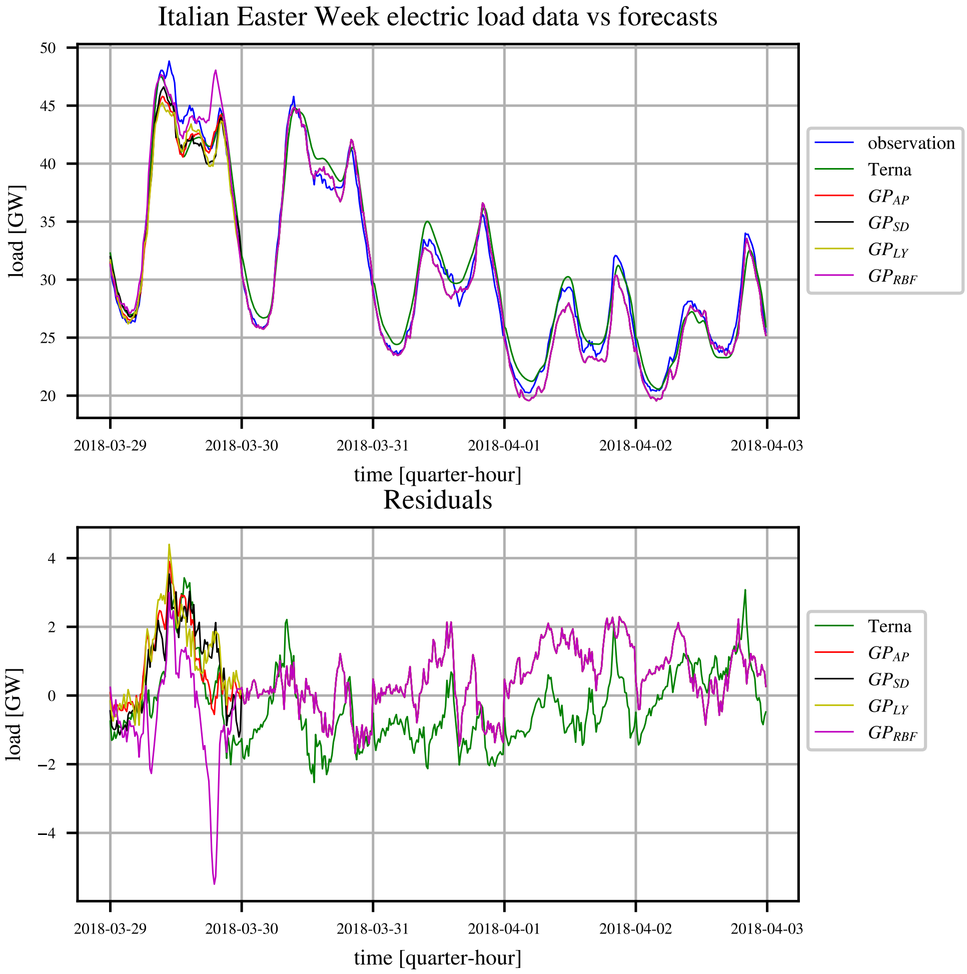


FIGURE 6.8: Easter Week of 2018: data vs forecasts and residuals.

TABLE 6.3: Year 2019: forecasting performances and percentage variation with respect to Terna (between brackets).

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[GW]	$\mathbf{MAE_{daily}}$[GW]
Terna	3.43	1.16	0.96	2.66	0.91	0.76
$\mathbf{GP_{AP}}$	3.02 (−12%)	1.23 (6%)	0.9 (−6%)	2.59 (−3%)	1.07 (18%)	0.76 (0%)
$\mathbf{GP_{SD}}$	2.83 (−17%)	1.07 (−8%)	0.82 (−15%)	2.37 (−11%)	0.92 (1%)	0.68 (−11%)
$\mathbf{GP_{LY}}$	3.35 (−2%)	1.57 (35%)	1.04 (8%)	2.98 (12%)	1.35 (48%)	0.9 (18%)
$\mathbf{GP_{RBF}}$	**2.68** (−22%)	**1.06** (−9%)	**0.79** (−18%)	**2.28** (−14%)	**0.86** (−5%)	**0.64** (−16%)

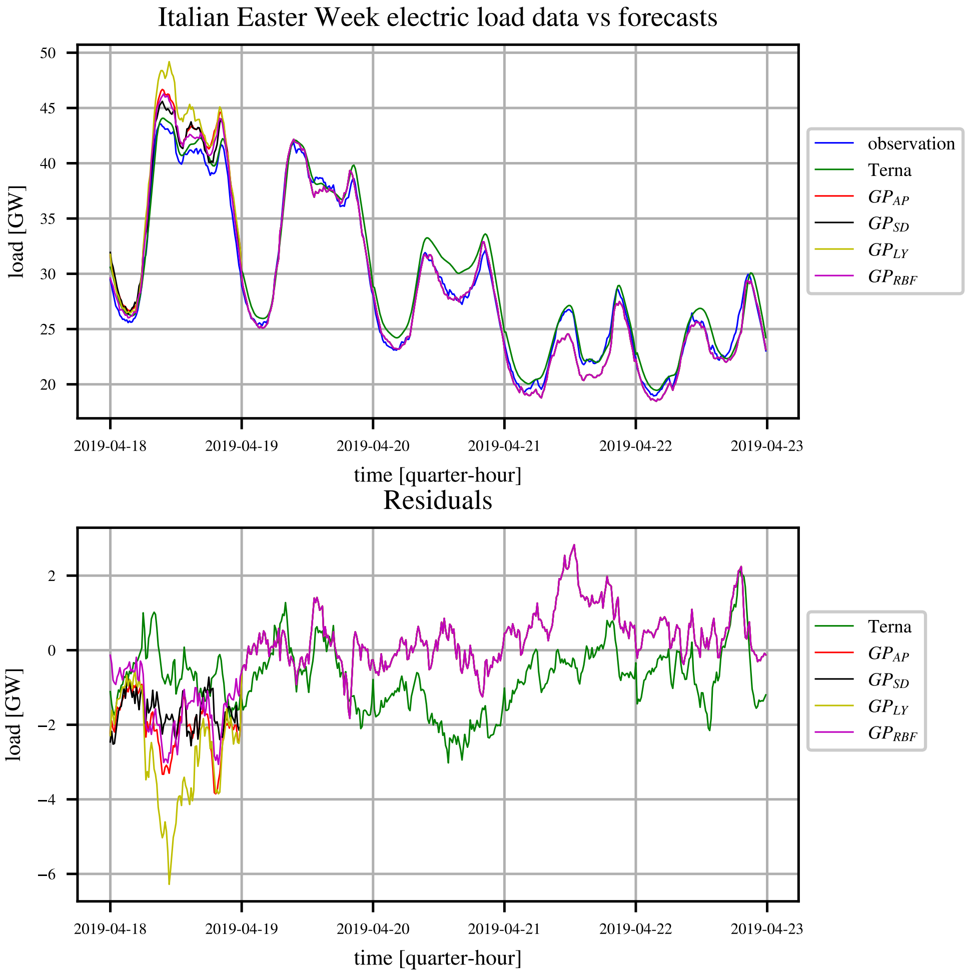


FIGURE 6.9: Easter Week of 2019: data vs forecasts and residuals.

They show a substantial improvement in all the considered indexes (e.g. up to 40% for quarter-hourly RMSE and 62% for daily MAE in 2017), with the $Avg(GP_{RBF})$ excelling in most of the cases.

6.6 Discussion

The problem of short-term prediction of the quarter-hourly electric load demand of Italy during the Easter Week has been addressed by means of a predictor based on a Gaussian Process model that ‘tracks’ the load demand shift with respect to a typical Easter Week profile. A ‘custom’ kernel, based on the sample autocovariance of the Easter Week profiles collected through three decades, was used to encode the prior information. Four alternative techniques were experimented for predicting the quarter-hourly profile of Thursday, the first day of the Easter Week. For this task, three long-term methods, namely ‘Average Profile’, ‘Similar Day’ and ‘Last Year’, and a short-term one, namely a Radial Basis function network trained on normal days, were considered.

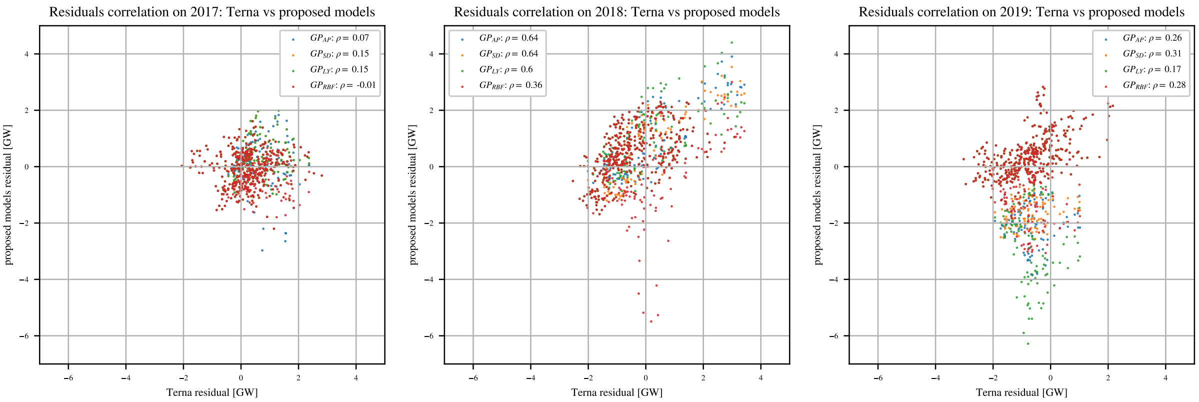


FIGURE 6.10: Scatter plots between Terna residual and proposed model residuals on the three test scenarios. The Pearson correlation coefficients indicate a weak correlation between the proposed models errors and Terna forecast error.

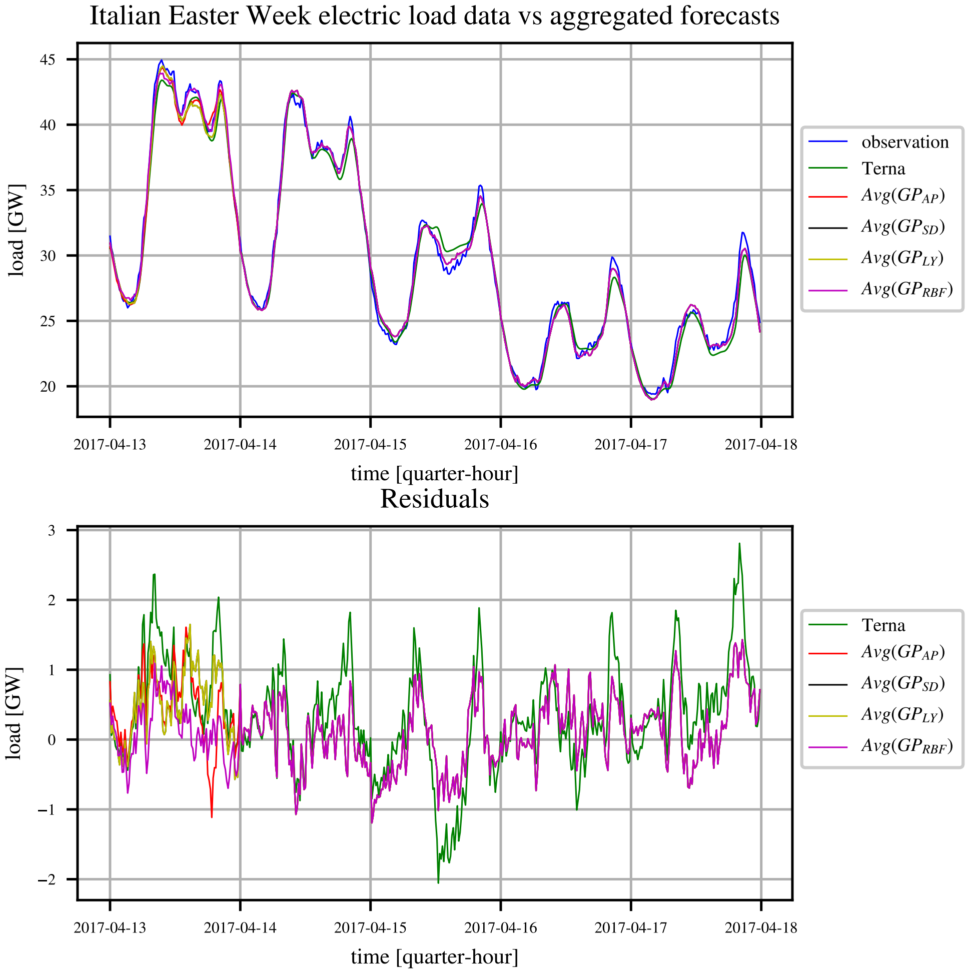


FIGURE 6.11: Easter Week of 2017: data vs aggregated forecasts and residuals.

TABLE 6.4: Year 2017: aggregated forecasts performances and percentage variation with respect to Terna (between brackets).

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[GW]	$\mathbf{MAE_{daily}}$[GW]
Terna	2.09	0.84	0.63	1.53	0.49	0.45
$\mathbf{Avg(GP_{AP})}$	1.51 (−28%)	0.55 (−35%)	0.45 (−28%)	0.83 (−46%)	0.27 (−45%)	0.24 (−47%)
$\mathbf{Avg(GP_{SD})}$	1.53 (−27%)	0.56 (−33%)	0.45 (−28%)	0.86 (−44%)	0.29 (−41%)	0.25 (−44%)
$\mathbf{Avg(GP_{LY})}$	1.53 (−27%)	0.56 (−33%)	0.45 (−28%)	0.86 (−44%)	0.29 (−41%)	0.25 (−44%)
$\mathbf{Avg(GP_{RBF})}$	**1.41** (**−33%**)	**0.5** (**−40%**)	**0.4** (**−37%**)	**0.66** (**−56%**)	**0.2** (**−59%**)	**0.17** (**−62%**)

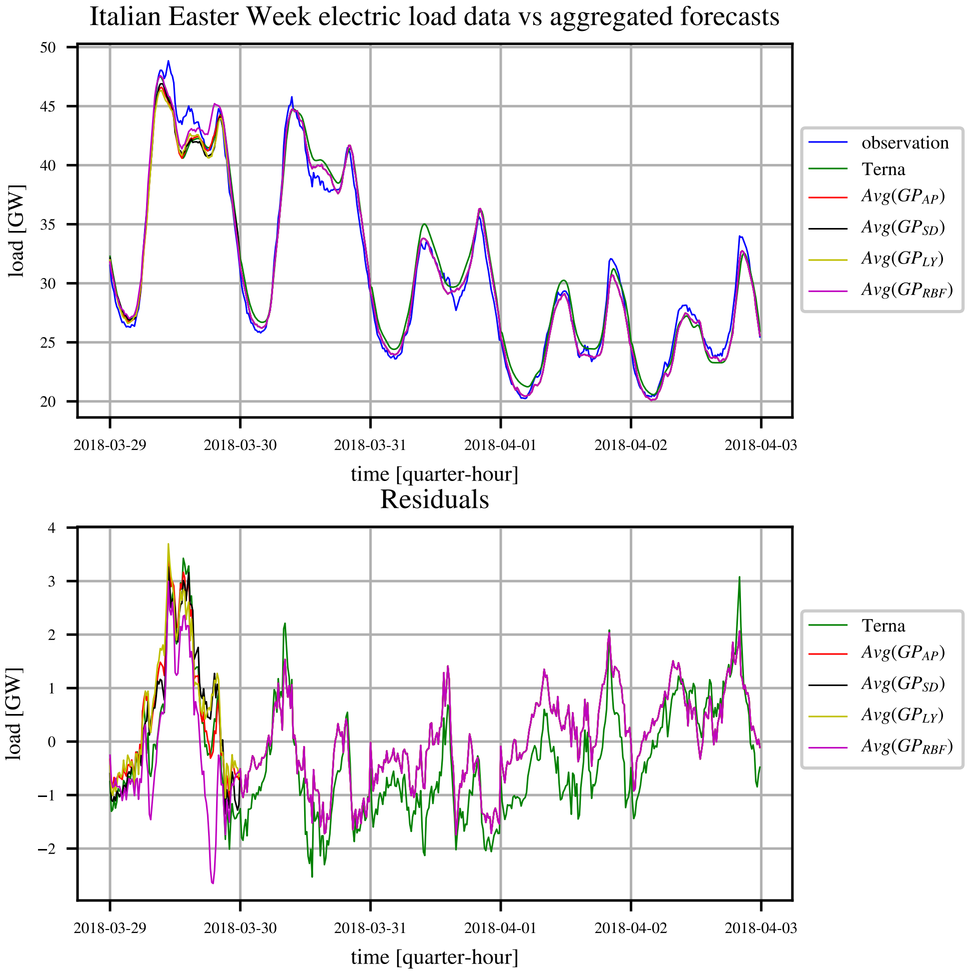


FIGURE 6.12: Easter Week of 2018: data vs aggregated forecasts and residuals.

Table 6.5: Year 2018: aggregated forecasts performances and percentage variation with respect to Terna (between brackets).

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	**MAPE**$_{\textbf{daily}}$[%]	**RMSE**$_{\textbf{daily}}$[GW]	**MAE**$_{\textbf{daily}}$[GW]
Terna	3.2	1.18	0.99	2.06	0.68	0.62
Avg(GP$_{\textbf{AP}}$**)**	**2.35** (**−27**%)	0.97 (−18%)	**0.75** (**−24**%)	1.69 (−19%)	0.52 (−25%)	0.51 (−19%)
Avg(GP$_{\textbf{SD}}$**)**	2.42 (−24%)	0.98 (−17%)	0.77 (−22%)	1.67 (−19%)	0.51 (−25%)	0.5 (−19%)
Avg(GP$_{\textbf{LY}}$**)**	2.37 (−26%)	0.97 (−18%)	0.76 (−23%)	1.74 (−16%)	0.54 (−21%)	0.53 (−15%)
Avg(GP$_{\textbf{RBF}}$**)**	2.4 (−25%)	**0.94** (**−20**%)	0.75 (−24%)	**1.45** (**−30**%)	**0.44** (**−35**%)	**0.42** (**−32**%)

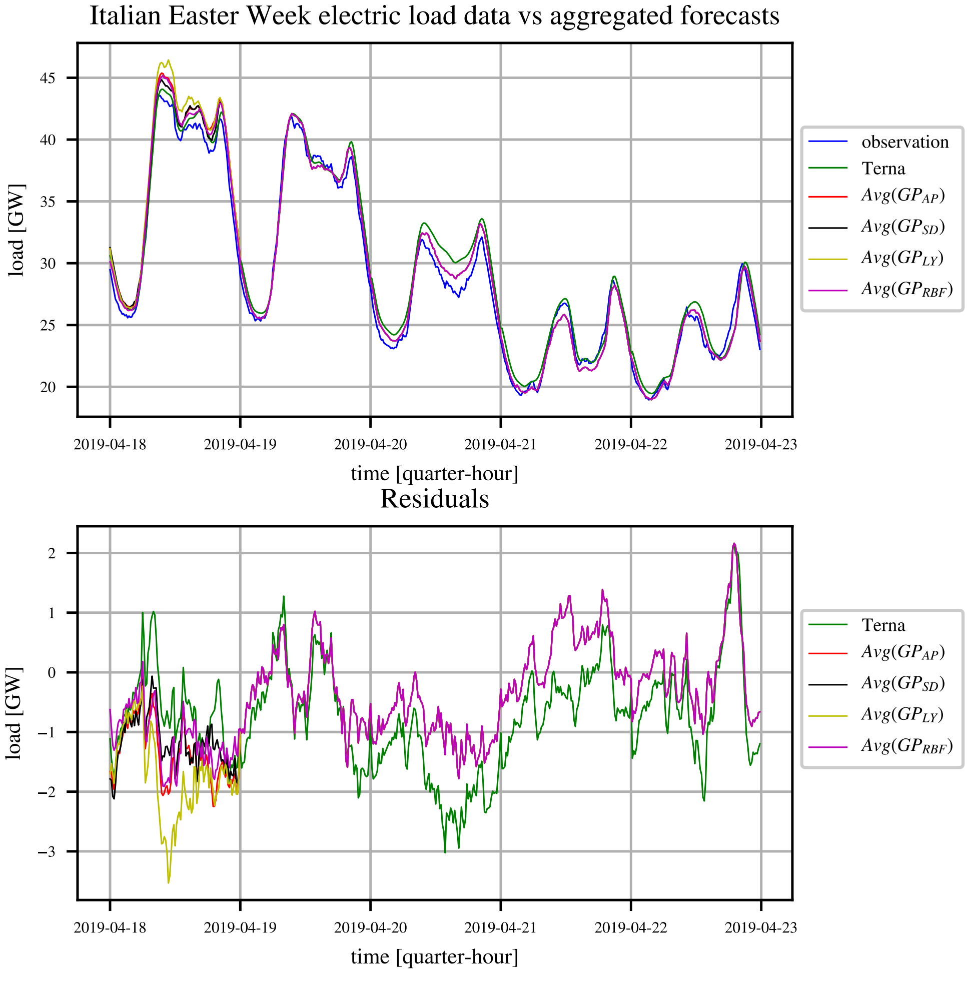


Figure 6.13: Easter Week of 2019: data vs aggregated forecasts and residuals.

Table 6.6: Year 2019: aggregated forecasts performances and percentage variation with respect to Terna (between brackets).

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[**GW**]	$\mathbf{MAE_{daily}}$[**GW**]
Terna	3.43	1.16	0.96	2.66	0.91	0.76
Avg($\mathbf{GP_{AP}}$)	2.62 (−24%)	0.95 (−18%)	0.78 (−19%)	1.88 (−28%)	0.77 (−15%)	0.58 (−24%)
Avg($\mathbf{GP_{SD}}$)	2.53 (−26%)	0.88 (−24%)	0.74 (−23%)	1.77 (−33%)	0.7 (−23%)	0.54 (−28%)
Avg($\mathbf{GP_{LY}}$)	2.79 (−19%)	1.08 (−7%)	0.85 (−11%)	2.08 (−22%)	0.9 (−1%)	0.65 (−14%)
Avg($\mathbf{GP_{RBF}}$)	**2.46** (**−28%**)	**0.87** (**−25%**)	**0.72** (**−25%**)	**1.73** (**−35%**)	**0.67** (**−26%**)	**0.52** (**−32%**)

The proposed strategies, tested over 2017-2019 yielded a substantial improvement in terms of MAPE, RMSE and MAE on both the quarter-hourly and the daily framework with respect to the benchmark predictor, i.e. the TSO one. The uncorrelatedness between the residuals of our and Terna forecasters suggested that a further improvement was achievable by a simple aggregation of the predictions, which indeed brought to a more than 40% drop in the main error indexes.

7

Aggregation of nonlinearly enhanced experts with application to electricity load forecasting

In this chapter, we consider the problem of obtaining a reliable short term load forecast, based on the availability of several ready-to-use forecasts from external providers, herein called 'base experts' [100] [101]. In view of their ready-to-use nature and the impossibility of retraining them, the expert forecasts are called 'secondary data'. By contrast, the features (predictors) associated with the target to be forecast are called 'primary data'. It is to be noted that in the case of time series, certain kinds of primary data are always available, since some features, such as daytime, weekday, and day of the year are typically known. This means that, in alternative to the simple aggregation of the secondary data, taken on an 'as is' basis, a more ambitious strategy can be pursued, i.e. the enhancement of the base forecasts by judicious use of the primary data.

The purpose of this chapter is to introduce a general aggregation scheme that uses the primary data in order to enhance the secondary ones, i.e. the expert forecasts, so as to fix their weaknesses and achieve a better final aggregated prediction. The learning framework assumes the availability of a training dataset made of primary and secondary data. The primary data include target and feature variables, while the secondary ones gather predictions made by external experts. It is worth remarking that the features available in the primary data may cover only partially those used by the experts to produce their predictions. For this reason, it is not convenient to drop the experts and retrain a model based on the primary data alone, because valuable information incorporated in the experts may go lost.

Two approaches are considered for blending the experts with the primary variables. The first one, called aggregation of enhanced experts (AEE), is a two-step approach. First, each expert is individually enhanced, accounting for the primary data. In the second step, the enhanced experts are aggregated by weighted average techniques. The second approach, conversely, consists of the enhanced aggregation of experts (EAE) through an Artificial Neural Network trained using primary and secondary data in order to predict the target. The possible occurrence of missing experts is addressed by a statistical imputation technique.

The short-term prediction of the German electrical load is used as a case study. In this context, the secondary variables to be aggregated are the predictions of twelve experts, whose underlying models are supposed to be unknown, while the primary variables are the calendar features, i.e. daytime (quarter-hourly measured) and weekday. The comparison of the two enhanced aggregation approaches to state-of-art aggregation methods shows that enhancement brings a significant improvement.

The chapter is organized as follows: in Section 7.1, the benchmark dataset is described. In Section 7.2 the *Expert enhancement/aggregation problem* is formulated and the two enhancement/aggregation strategies are described. In Section 7.3, the different methods are applied to the benchmark data in both 'full information' and 'missing-experts' scenario and the results are shown. In Section 7.4, the main conclusions are summarized and discussed.

7.1 Benchmark problem: German electricity load forecasting

Consider the forecasting of the country electric load of Germany, denoted by

$$y(t_k) \in \mathbb{R}, \quad t_k - t_{k-1} = 0.25 \text{ hour}, \forall k$$

For the sake of simplicity, hereafter the shorthand notation $y(k)$ will be used in place of $y(t_k)$.

The actual load demand, recorded from January 1 to September 6, 2019, is displayed in Fig. 7.1. The quarter-hourly data (MW) were downloaded from the Entso-E transparency platform [102].

During this period, the forecasts provided by twelve experts, trained over the years 2017 and 2018, are available:

$$\xi_i(k) \in \mathbb{R}, \quad i \in \mathcal{M}, \quad \mathcal{M} = \{1, \ldots, 12\}$$

All these experts are underpinned by models belonging to the Generalized Additive Models (GAMs) category [103], but they differ either in the choice of the features or the calibration procedures. In the literature, the term 'sister forecasts' has been used to denote families of forecasters sharing a common structure [9]. The features considered by each expert are summarized in Table 7.1. Among them, there is also the binary d_{type} feature that identifies special days such

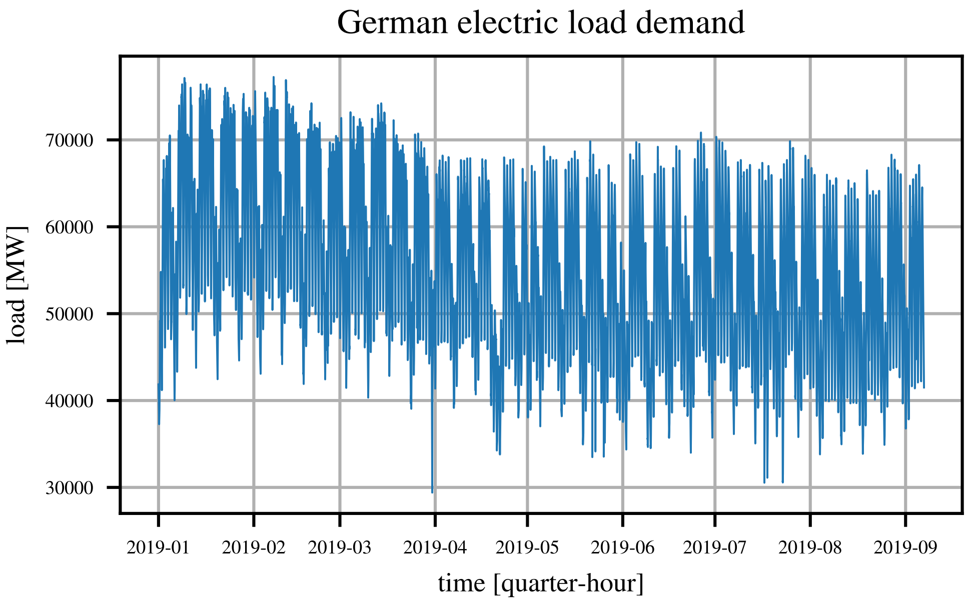


FIGURE 7.1: German quarter-hourly electric load demand from January 1^{st} to September 6^{th}, 2019.

as holidays. For simplicity of notation, from now on we will refer to the 'experts forecasts' just as 'experts'.

The model structures underpinning the experts are supposed to be unknown which means that their possible weaknesses are not known *a priori* but can nevertheless be assessed *ex post* from the observed data. A preliminary exploration can be performed by inspecting the prediction residuals $r_i := y - \xi_i$. In particular, the joint bivariate distributions of the residuals for all possible pairs of experts can be visualized through the scatter matrix displayed in Fig. 7.2, where also the correlation coefficients are reported. There are pairs of highly correlated residuals (e.g. experts 2 and 3) and other pairs whose correlation is poor (e.g. experts 3 and 11). Of course, if all expert residuals were highly correlated, there would be no room for improvement by aggregation.

The correlations between residuals and observations are shown in red in the last column of Fig. 7.2 and the correlation coefficients are shown in the last row. A highly performing expert will be characterized by low correlation and small residual variance, see e.g. experts 9 and 11, whose scatter plots appear horizontal and thin. Conversely, a significant correlation is not only a symptom of suboptimality, but could be leveraged to reduce the error, a notable example being represented by expert 12.

The twelve univariate residual distributions can be compared through their boxplots, see Fig. 7.3. It appears that in all cases the residuals are negatively biased, meaning that in the considered period all the experts tend to overestimate the load.

Table 7.1: Features of the expert models. T and I represent respectively the temperature and the normal solar radiation, t_{year} and t_{day} are calendar features associated with the time of year and the time of day, and d_{type} is the binary feature identifying special days such as holidays.

Expert	**Features**
Expert 1	$t_{year}, t_{day},\ T(k),\ \xi(k-24),\ d_{type}$
Expert 2	$t_{year}, t_{day},\ T(k),\ \xi(k-24),\ d_{type}$
Expert 3	$t_{year}, t_{day},\ T(k),\ \xi(k-24),\ I(k),\ d_{type}$
Expert 4	$t_{year}, t_{day},\ I(k),\ T(k-8),\ T(k-16),$ $T(k-24),\ \xi(k-8), \xi(k-16),\ \xi(k-24),\ d_{type}$
Expert 5	$t_{year}, t_{day},\ I(k),\ T(k-8),\ T(k-16),$ $T(k-24),\ \xi(k-8), \xi(k-16),\ \xi(k-24),\ d_{type}$
Expert 6	$t_{year}, t_{day},\ I(k),\ T(k),\ T(k-1),\ T(k-2),$ $T(k-3), T(k-4),\ T(k-24),\ \xi(k-24),\ d_{type}$
Expert 7	$t_{year}, t_{day},\ I(k),\ T(k-24),$ $T(k-48),\ \xi(k-24),\ \xi(k-48),\ d_{type}$
Expert 8	$t_{year}, t_{day},\ I(k),\ T(k),\ T(k-8),\ T(k-16),$ $T(k-24),\ \xi(k-12),\ \xi(k-16),\ \xi(k-24),\ d_{type}$
Expert 9	$t_{year}, t_{day},\ I(k),\ T(k),\ T(k-4),\ T(k-8),$ $T(k-12),\ T(k-16),\ T(k-20),$ $T(k-24), \xi(k-8),\ \xi(k-16),\ \xi(k-24),\ d_{type}$
Expert 10	$T(k),\ T(k-4),\ T(k-8),\ T(k-12),\ T(k-16),$ $T(k-20),\ T(k-24), \xi(k-12),\ \xi(k-24),\ d_{type}$
Expert 11	$I(k),\ T(k),\ T(k-4),\ T(k-8),\ T(k-12),$ $T(k-16),\ T(k-20),\ T(k-24), \xi(k-8),$ $\xi(k-8), \xi(k-12),\ \xi(k-24),\ d_{type}$
Expert 12	$T(k-24),\ T(k-48),\ T(k-72),$ $\xi(k-24), \xi(k-48),\ \xi(k-72),\ d_{type}$

A visual example of the weaknesses of some of the experts is given in Fig. 7.4, where experts 1 and 10 are plotted over a sample week together with the observations. Both experts fail to catch the load demand profile during the night, expert 10 overestimates the morning demand during the weekend and expert 1 exhibits a daily pattern which is way too smooth with respect to the actual shape of the load demand. A more systematic insight can be gained by looking at Fig. 7.5, reporting the scatter plots of the quarter-hour of the day against the residuals of experts 1 and 10 on Mondays and Saturdays. The consistent patterns seen in these scatter plots are not specific to these two experts, but analogous patterns, depending on the day of the week, are found also in the residuals of the other experts.

From this exploratory analysis it emerges that there exists room for improvement, provided that the the weaknesses of the experts are fixed by a suitable enhancement technique. These

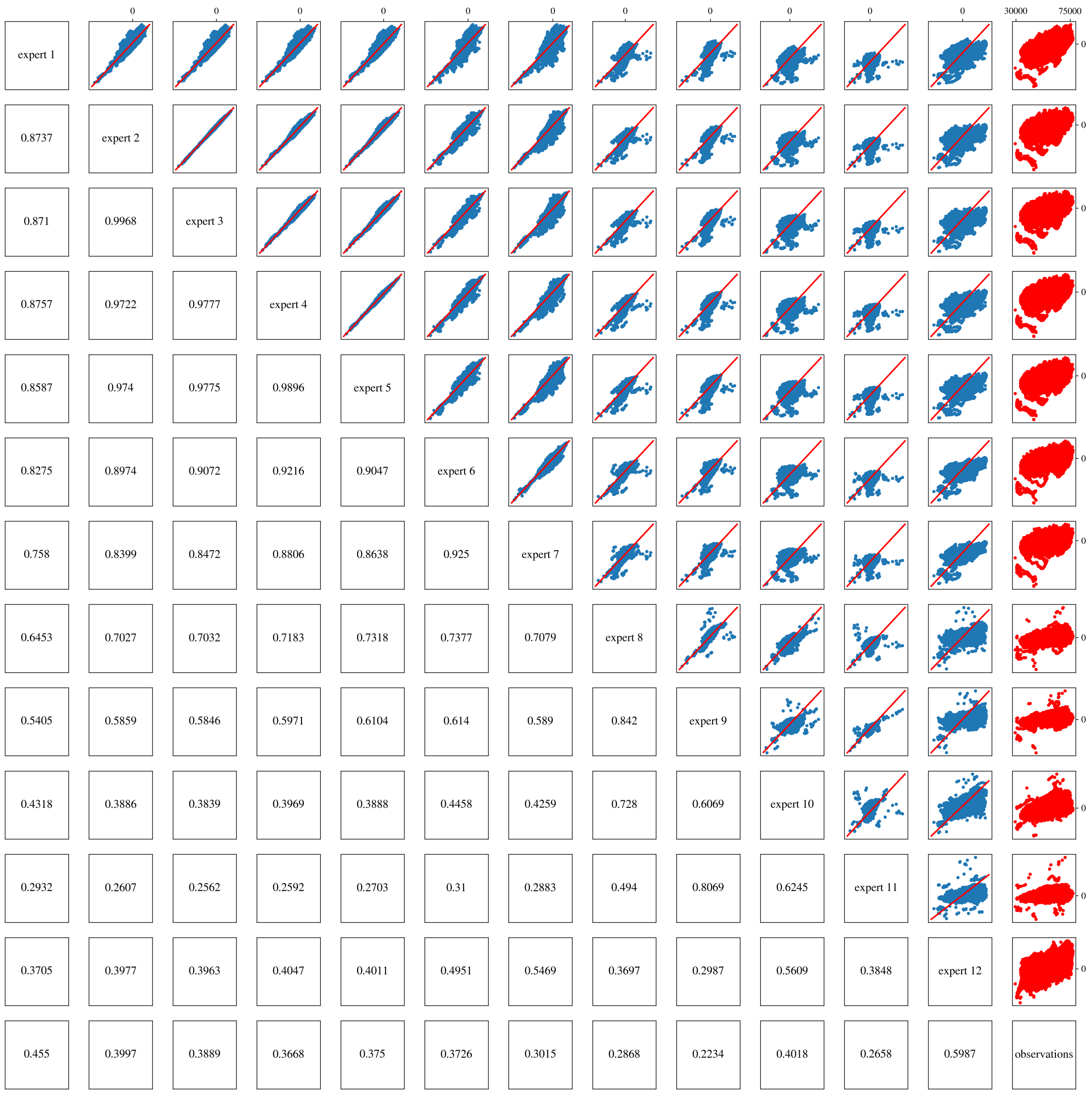


FIGURE 7.2: Forecasting residuals of the 12 experts and observed German loads: scatter matrix and correlation coefficients.

weaknesses are not necessarily due to inadequate design of the experts. In fact, the distribution of the observations could have changed over time. In particular, the correlation between some features and the target might have changed.

The exploratory analysis highlighted also a possible way to achieve an improvement, since the residuals are correlated with calendar features such as the weekday and the quarter-hour of the day, that can be regarded as exogenous features, usable to enhance the base experts.

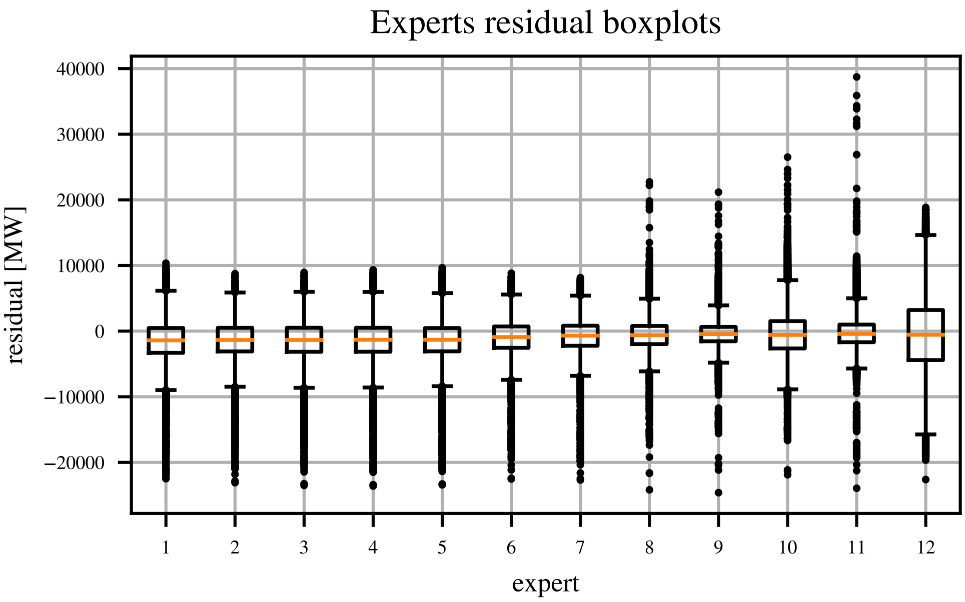


FIGURE 7.3: Boxplots of the forecasting residuals of the 12 experts.

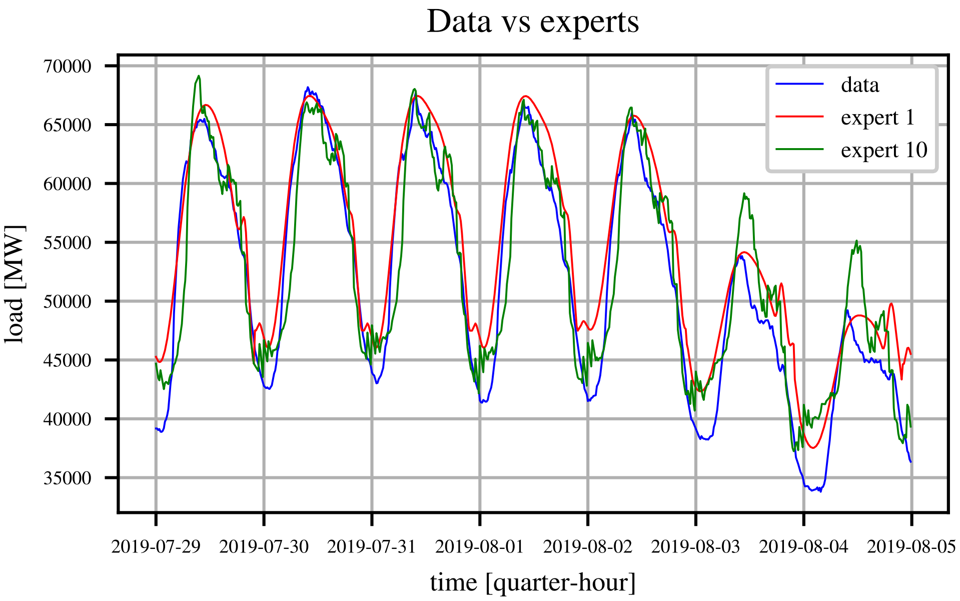


FIGURE 7.4: Observed German loads (blue), Expert 1 forecasts (red), and Expert 10 forecasts (green) during a one week period.

7.2 Expert enhancement and aggregation

7.2.1 Problem statement

Motivated by the German load forecasting example, we are now in a position to state the *Expert enhancement/aggregation problem.* In the following, $\mathcal{T}r$ and $\mathcal{T}e$ will denote the set of time indices associated to the training and test set, respectively.

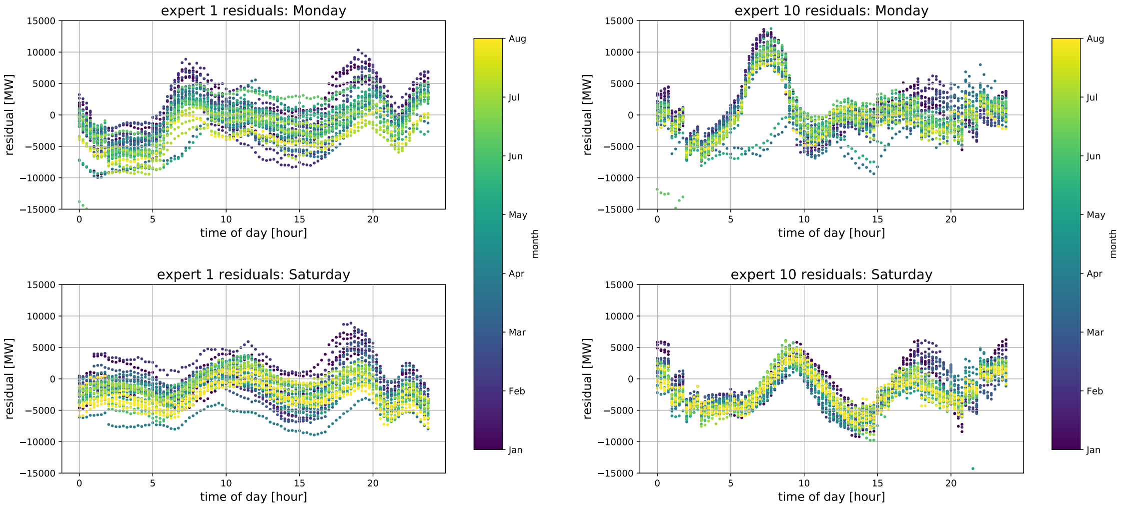


FIGURE 7.5: Scatter plots of forecasting residuals of Expert 1 (left panels) and 10 (right panels) against time of day on Monday (top panels) and Saturday (bottom panels).

Problem Statement. Given the training data $\{y(k), \boldsymbol{\xi}(k), \boldsymbol{x}(k)\}$, $k \in \mathcal{Tr}$, where $\boldsymbol{\xi}(k) = [\xi_1(k) \dots \xi_m(k)]^T$ is the vector of the experts and $\boldsymbol{x}(k) = [x_1(k) \dots x_p(k)]^T$ are exogenous features, devise an *expert enhancer* $H(\cdot)$ that provides an estimate $\hat{y} = H(\boldsymbol{\xi}, \boldsymbol{x})$ of the target variable y.

In the German load problem, this amounts to develop a forecaster $\hat{y}$ blending the contributions of the twelve experts ξ_i, $i = 1, \dots, 12$, with some exogenous features x_i, $i = 1, 2$, that, in the case considered, are chosen as the quarter-hour of day and the weekday.

The training and the testing steps require suitable datasets. The whole available dataset does not cover an entire year and using consecutive periods of time for the training and testing phases could lead to misleading results because of the yearly seasonality. This motivated the adoption of a random selection of training days and testing days. In particular, one third of the entire dataset is reserved for testing purposes.

Typical performance metrics are the Mean Absolute Percentage Error

$$MAPE = \frac{100}{n} \sum_{k \in \mathcal{Te}} \left| \frac{\hat{y}(k) - y(k)}{y(k)} \right|$$

and the Root Mean Square Error

$$RMSE = \sqrt{\frac{\sum_{k \in \mathcal{Te}} (\hat{y}(k) - y(k))^2}{n}}$$

In the following, we propose two approaches for the *Expert enhancement/aggregation problem*: the aggregation of enhanced experts (AEE) and the enhanced aggregation of experts (EAE).

7.2.2 Aggregation of enhanced experts (AEE)

Individual enhancers

This approach splits the enhanced aggregation process in two phases. First, each expert is fed into an individual enhancer $h_i(\cdot,\cdot)$ that, exploiting the exogenous features, yields the enhanced expert

$$\hat{\xi}_i = h_i(\xi_i, \boldsymbol{x})$$

Then, the enhanced experts undergo a final aggregation step that yields the final prediction

$$\hat{y} = G(\hat{\xi})$$

where $G(\cdot)$ is the aggregator function and $\hat{\boldsymbol{\xi}} = [\hat{\xi}_1 \dots \hat{\xi}_m]$ is the vector of the enhanced experts. Different strategies can be adopted for the design of the individual enhancers. Since the basic idea is a sort of 'retuning' of the expert, based on few exogenous features, the enhancer should privilege simplicity and robustness. The simplest enhancers are the additive, multiplicative and affine ones. Since they implement basic correction schemes they would be the first to be included in the 'enhancers toolbox' used to compute the enhanced experts that will go over to the second step, i.e. aggregation. In many contexts, one could take advantage of more complex enhancers, able to model arbitrary nonlinear correction schemes. Two examples are the regularization-based tensor spline and the MLP Artificial Neural Network. Of course other nonlinear models could be used to enrich the suite of possible enhancers. Below, five enhancers are considered: additive, multiplicative, affine, tensor spline and MLP ANN.

Additive Enhancer. A simple enhancement consists of an additive correction term β_i that depends nonlinearly on the exogenous features:

$$\hat{\xi}_i = \xi_i + \beta_i(\boldsymbol{x}), \quad i \in \mathcal{M}$$

For the benchmark problem,

$$\hat{\xi}_i = \xi_i + \beta_i(t_{day}, d_{week}), \quad i = 1, ..., 12$$

where the weekday $d_{week} \in \{\text{Monday}, \dots, \text{Sunday}\}$ is a categorical variable and the quarter-hour time t_{day}, $1 \leq t_{day} \leq 96$, is treated as a real variable on the circle. For this reason, training $\beta(\cdot,\cdot)$ amounts to training seven periodic longitudinal functions $\beta_i(\cdot, d_{week})$, $d_{week} \in$

$\{\text{Monday}, \ldots, \text{Sunday}\}$. In order to account for the 24-hour periodicity, cyclic penalized cubic B-splines (cyclic P-splines) were adopted as basis functions. The squared second derivative was used as regularization penalty. With this choice, the regularization hyperparameter determines the rate of change of the correction term throughout the 24 hours.

Multiplicative Enhancer. Another type of enhancement consists of a multiplicative correction term α_i that depends nonlinearly on the exogenous features:

$$\hat{\xi}_i = \xi_i \alpha_i(\boldsymbol{x})$$

For the benchmark, this translates into the model

$$\hat{\xi}_i = \xi_i \alpha_i(t_{day}, d_{week}) \quad i = 1, ..., 12$$

The structure and training of α_i is analogous to that of β_i in the additive enhancer.

Affine Enhancer. When both additive and multiplicative correction terms are employed, we obtain the affine enhancer:

$$\hat{\xi}_i = \xi_i + \beta_i(\boldsymbol{x}) + \xi_i \alpha_i(\boldsymbol{x})$$

It is worth observing that the enhancer has a GAM structure, meaning that a backfitting iteration [103] [104] can be used for its training: at each step, only the estimation of either α_i and β_i is performed. For the benchmark, the affine enhancer takes the form

$$\hat{\xi}_i = \xi_i + \beta_i(t_{day}, d_{week}) + \xi_i \alpha_i(t_{day}, d_{week})$$

The structures of α_i and β_i remain the same as in the additive and multiplicative enhancers. By resorting to backfitting, at each step only the estimation of functions of the quarter-hour time is required.

Tensor Spline Enhancer. A more flexible enhancer is obtained by resorting to a generic nonlinear function described by a tensor spline model:

$$\hat{\xi}_i = \xi_i + f_i^{\text{tensor}}(\xi_i, \boldsymbol{x})$$

For the benchmark,

$$\hat{\xi}_i = \xi_i + f_i^{\text{tensor}}(\xi_i, t_{day}, d_{week})$$

where $f_i^{tensor}(\cdot, \cdot, d_{week})$, $d_{week} \in \{\text{Monday}, \ldots, \text{Sunday}\}$ are seven cubic bivariate tensor splines, describing a function of two variables (expert and the quarter-hourly time of day). The tensor functions are built using P-splines for the expert component direction and cyclic P-splines for the quarter-hour time of day.

MLP-ANN Enhancer. Another flexible enhancer uses as correction term a Multi-Layer Perceptron Artificial Neural Network (MLP-ANN) f^{MLP} with input vector $[\xi_i \quad \boldsymbol{x}^T]^T$:

$$\hat{\xi}_i = \xi_i + f_i^{\mathrm{MLP}}([\xi_i \quad \boldsymbol{x}^T]^T)$$

For the benchmark, we let $x = t_{day}$ and seven ANN's are trained, one for each weekday.

Calibration of the hyperparameters

The first four enhancers belong to the Generalized Additive Models (GAMs) family and were implemented using the `Python`'s library `pyGAM`. The Artificial Neural Network was implemented using the `Python`'s library `scikit-learn`.

For what concerns the splines terms of the GAMs-based enhancements, a fairly large number of basis functions was used, entrusting the regularization task to the penalty hyperparameter. In particular, 12 cyclic P-splines were used for the daily periodicity (one spline every two hours) and 20 P-splines for the expert prediction variable.

The regularization parameters were tuned, for each expert, using Generalized Cross Validation (GCV).

All the Neural Networks considered in this chapter were one-hidden-layer networks with 100 neurons in the hidden layer. This value was chosen through trial and error approach. The model parameters were computed through backpropagation using a penalized squared loss function, where the penalization consists of an L2-regularization term. Optimization was carried out by the `adam` stochastic gradient-based optimizer proposed by [105]. The regularization parameter was tuned through K-fold cross-validation.

Experts aggregation

The second and final step of the AEE scheme is the aggregation of the enhanced predictors. In the following, five possible aggregation methods are reviewed.

Simple average. Despite its simplicity, the arithmetic mean is a widely used method for aggregating individual experts. For some economic time series it was even found that only few aggregation schemes outperformed the forecast obtained by means of experts [106]. The simple average is a robust method and deals seamlessly with the problem of missing experts.

Winsorized average. The Winsorized average (WA) is a more robust version of the simple average method, where the two extreme experts are replaced by the second largest and the second smallest experts.

Trimmed average. The Trimmed average (TA) is another robust extension of the simple average method, where at each time the two extreme experts are discarded from the computation.

Constrained Least Squares. The Constrained Least Squares (CLS) average is a weighted linear combination, whose weights are chosen so as to minimize the sum of squared residuals, constraining the weights to be nonnegative and to sum up to one. More precisely,

$$\hat{y} = \boldsymbol{\theta^T} \hat{\boldsymbol{\xi}}, \quad \hat{\boldsymbol{\xi}} = \left[\hat{\xi}_1 \dots \hat{\xi}_m\right]^T$$

$$\begin{aligned} \theta = \arg\min_{\theta} \sum_{k \in \mathcal{T}r} & \left(y(k) - \boldsymbol{\theta}^T \hat{\xi}(k)\right)^2 \\ \text{s.t.} \quad & \theta_i \geq 0, \quad \forall i \\ \sum_{i \in \mathcal{M}} & \theta_i = 1 \end{aligned}$$

This has not only the advantage of preventing weights instability that may ensue from the the collinearity of the experts, but guarantees a more interpretable model as well [9]. For the benchmark problem, the optimal weights were computed through quadratic programming, using the `Python`'s library `cvxopt`.

Multi-Layer Perceptron. In order to assess the potential of nonlinear aggregation, a one-hidden layer MLP-ANN aggregation method is considered as well:

$$\hat{y} = f^{\text{MLP}}(\hat{\boldsymbol{\xi}})$$

Structure and training of the network are analogous to those of the individual MLP-ANN enhancer described in 7.2.2.

7.2.3 Enhanced Aggregation of Experts (EAE)

The second approach, enhanced aggregation of experts, is a one-step aggregation scheme that feeds directly the experts and the exogenous features to a unique nonlinear enhancer:

$$\hat{y} = h(\boldsymbol{\xi}, \boldsymbol{x})$$

Of course, a variety of choices is possible for describing the nonlinear relationship $h(\cdot,\cdot)$. Here, an MLP-ANN is used with structure and training analogous to those of 7.2.2.

For the benchmark problem, seven MLP-ANN's were trained, one for each weekday, fed by the vector $\boldsymbol{\xi} \in \mathbb{R}^{12}$ of experts predictions and the quarter-hour of the day t_{day}:

$$\hat{y} = h_{\text{weekday}}(\boldsymbol{\xi}, t_{day}), \quad \text{weekday} \in \{\text{Monday}, \dots, \text{Sunday}\}$$

7.3 Results

In this Section the two proposed approaches, AEE and EAE, are applied to the German load benchmark. First the effects of the individual enhancers are presented. Subsequently, the enhanced experts are aggregated according to the AEE scheme and the results compared to those of the direct EAE scheme.

7.3.1 Individual enhancement

For the benchmark problem, each individual enhancer can be visualized as seven surfaces, one for each day. The surfaces return the enhanced expert prediction, i.e. the enhanced load forecast, as a function of the original expert prediction and t_{day}. For a given weekday, displaying these surfaces offers an effective visualization of the alternative individual enhancement methods. In particular, let us consider the five enhancers discussed in section 7.2.2. For expert 12, the five 'Thursday surfaces' are displayed in Fig. 7.6, together with the test data. The first plot is the surface corresponding to no enhancement, i.e. the 45-degree plane $\hat{\xi}_{12} = \xi_{12}$. In the insets, the corresponding Goodness of Fit plots are provided for the test data. The closer the surface is to the test data, the more effective is the enhancer. It can be seen that, without enhancement, $R^2 = 0.66$. All the five enhancers raise R^2 above 0.8, the maximum value being achieved by the affine enhancer ($R^2 = 0.87$).

The MAPE and RMSE of each expert on the test data is compared to the ones of its five enhanced versions in Table. 7.2 and Table. 7.3. It appears that in all cases, individual enhancement significantly improves the base experts. Although no enhancer is uniformly superior to the others, the affine enhancer ranks first in seven cases out of twelve for the MAPE and six cases out of twelve for the RMSE. The second best enhancer appears to be the tensor spline one, that ranks first in five cases out of twelve for both the MAPE and the RMSE.

The different flexibility of the alternative enhancers can be appreciated in Fig. 7.7, where three sections (at times 06:00, 06:30, and 07:00) of the five 'Tuesday surfaces' of the five enhancers are superimposed. In particular, it can be seen that the additive and multiplicative enhancers are

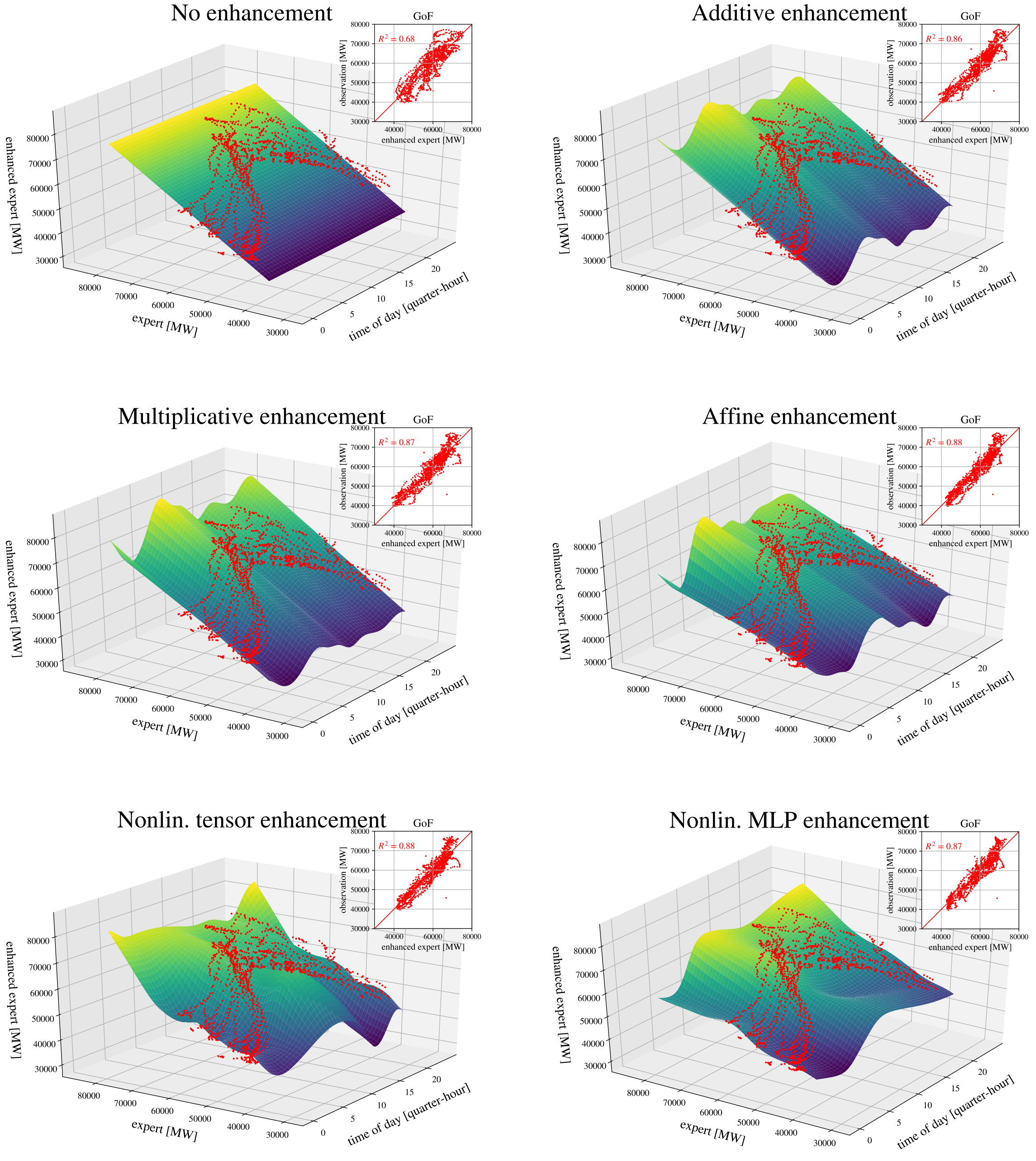


Figure 7.6: Expert 12: actual loads against expert's forecast and time of day. Test data: red; enhancement function: surface. Insets: Goodness-of-Fit plots for test data.

not flexible enough to follow the training and test data. Conversely, the other three enhancers offer a comparable performance.

7.3.2 AEE and EAE results

For the AEE approach, 25 results were obtained, by considering all the possible pairs given by one of the five individual enhancers (Additive, Multiplicative, Affine, Nonlin. tensor, Nonlin.

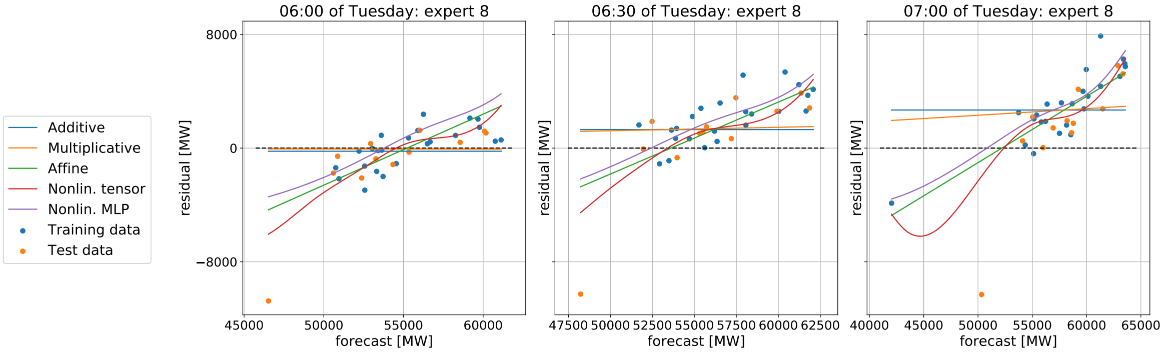


FIGURE 7.7: Expert 8: forecasting residuals against expert's forecast in correspondence of three times of day. Training data: blue, test data: red, enhancement functions: continuous lines.

TABLE 7.2: Test MAPE [%] of each expert with and without enhancements, with the percentage decrease with respect to the not-enhanced expert below the best results.

Expert	**Enhancing methods**					
	None	**Add.**	**Mul.**	**Aff.**	**Tensor**	**MLP**
Expert 1	5.24	3.74	3.78	**3.38** **(-35%)**	3.48	3.56
Expert 2	4.92	3.76	3.82	3.51	**3.41** **(-31%)**	3.53
Expert 3	4.96	3.84	3.89	3.59	**3.52** **(-30%)**	3.64
Expert 4	4.91	3.82	3.87	3.55	**3.45** **(-30%)**	3.52
Expert 5	4.91	3.83	3.88	3.56	**3.43** **(-30%)**	3.51
Expert 6	4.17	3.28	3.33	**2.99** **(-28%)**	3.02	3.14
Expert 7	3.83	3.15	3.18	3.07	**3.03** **(-21%)**	3.11
Expert 8	3.48	2.63	2.66	**2.44** **(-30%)**	2.52	2.51
Expert 9	2.80	2.14	2.16	**2.01** **(-28%)**	2.23	2.03
Expert 10	5.33	2.83	2.84	**2.72** **(-49%)**	2.86	3.10
Expert 11	3.54	2.52	2.53	**2.42** **(-32%)**	2.84	2.46
Expert 12	8.61	5.31	5.16	**4.68** **(-46%)**	4.74	4.90

MLP-ANN) associated with one of the five aggregation methods (simple, trimmed, Winsorized, CLS, MLP). Conversely, the EAE approach yields a single result.

Two testing scenarios were considered: a 'full information' scenario, where at each time instant all the twelve experts are available, and a 'missing-experts' one, where on some days only a subset of the experts is available.

Table 7.3: Test RMSE [MW] of each expert with and without enhancements, with the percentage decrease with respect to the not-enhanced expert below the best results.

Expert	**Enhancing methods**					
	None	**Add.**	**Mul.**	**Aff.**	**Nonlin**	**MLP**
Expert 1	3634	2869	2883	**2683** **(-26%)**	2894	2697
Expert 2	3416	2862	2888	2718	**2662** **(-22%)**	2714
Expert 3	3436	2899	2924	2759	**2731** **(-21%)**	2768
Expert 4	3448	2930	2956	2772	**2745** **(-20%)**	2737
Expert 5	3435	2940	2966	2786	**2709** **(-21%)**	2729
Expert 6	3009	2548	2567	**2373** **(-21%)**	2475	2442
Expert 7	2854	2535	2547	2467	**2449** **(-14%)**	2479
Expert 8	2541	2098	2098	**1942** **(-24%)**	2115	1964
Expert 9	2102	1754	1749	**1658** **(-21%)**	2049	1673
Expert 10	3869	2291	2268	**2205** **(-43%)**	2353	2304
Expert 11	2737	2117	2108	2029 **(-26%)**	3640	**1995** **(-27%)**
Expert 12	5752	3766	3703	**3529** **(-39%)**	3584	3614

Full information scenario

For the German load benchmark, the test MAPEs of the aggregated forecasts in the 'full information' scenario are summarized in Table 7.4. By a comparison with Table 7.2, the significant benefits brought by the two-stage strategies are apparent. While the original experts' MAPEs ranged from 2.80% to 8.61%, the AEE final MAPEs range from 1.95% to 2.86%, while the EAE scheme achieves a MAPE of 1.98%. Among the proposed methods, the Affine-CLS AEE method provides the best results, yielding a 30% improvement with respect to the best not-enhanced expert. Concerning AEE, it can be noted that: (i) inspection of the best expert column shows that individual enhancement plays a key role in the final performance because the best enhanced experts is already close to the optimal performance; (ii) the choice of the aggregation method is crucial, with CLS and MLP outperforming the other three simpler alternatives. The test RMSEs, reported in Table 7.5, can be compared with Table 7.3. Considerations are similar to those already made for the MAPE results.

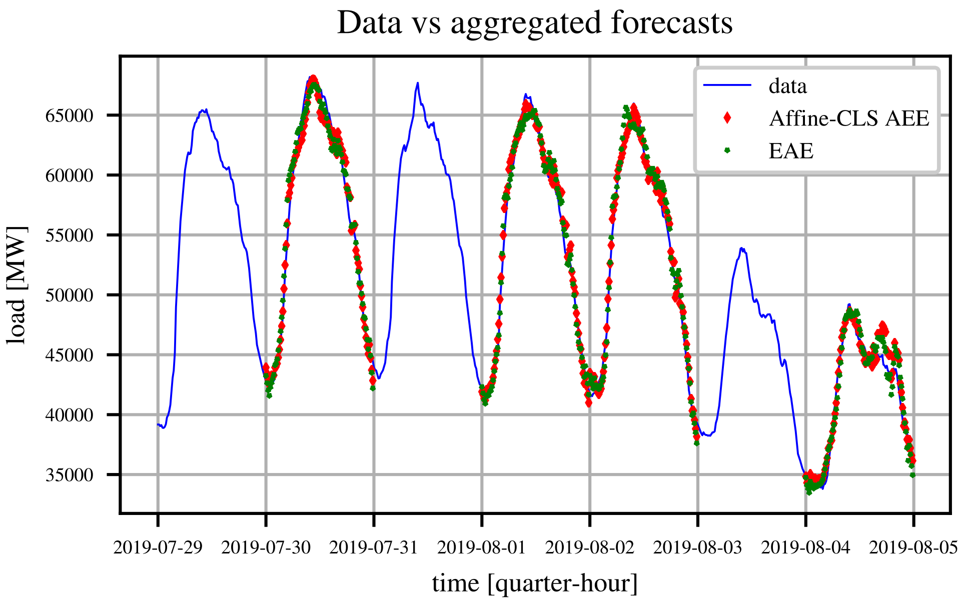


FIGURE 7.8: German load against time during one week. Observed loads: blue; Affine-CLS AEE: red; EAE: green.

TABLE 7.4: German data: test MAPE [%] of the proposed aggregation techniques. The best results are highlighted in bold; between parentheses, the percentage decrease with respect to the best not-enhanced expert (lower left corner).

		AEE					**EAE**
Individual Enhancement	**Best expert**	**Simple average**	**Trimm. average**	**Winsor. average**	**Constr. Least Squares**	**MLP**	
Additive	2.14	2.75	2.82	2.78	**2.09 (-25%)**	2.10	**1.98 (-29%)**
Multiplicative	2.16	2.79	2.86	2.82	**2.11 (-25%)**	2.15	
Affine	2.01	2.53	2.57	2.54	**1.95 (-30%)**	2.00	
Nonlin. tensor	2.23	2.50	2.51	2.48	**2.11 (-25%)**	2.20	
Nonlin. MLP-ANN	2.03	2.59	2.61	2.58	**1.97 (-30%)**	1.97	
No-enhancement	2.80	3.89	3.84	3.82	2.80	2.38	

Fig. 7.8 displays one week of data, plotting the predictions provided by the Affine-CLS AEE (red) and the EAE (green) during the four days (Tuesday, Thursday, Friday, Sunday) randomly chosen for testing. The comparison with Fig. 7.4, where the predictions of two base experts were displayed, shows that both aggregation strategies AEE and EAE are capable of providing accurate predictions of the testing data.

Missing-experts scenario

In the missing-experts scenario, one, multiple, or even all experts may be unavailable during some time windows. It is important to consider this case as well, because these occurrences are not rare, due to disruptions in communications and software or hardware failures [107]. While mean-based aggregation techniques can naturally deal with missing values, weighted and nonlinear aggregation methods cannot. Herein, the conditional mean imputation method is

Table 7.5: German data: test RMSE [MW] of the proposed aggregation techniques. The best results are highlighted in bold; between parentheses, the percentage decrease with respect to the best not-enhanced expert (lower left corner).

Individual Enhancement	**Best expert**	**AEE**					**EAE**
		Simple average	**Trimm. average**	**Winsor. average**	**Constr. Least Squares**	**MLP**	
Additive	1754	2179	2234	2203	**1714 (-18%)**	1725	**1651 (-21%)**
Multiplicative	1749	2190	2247	2213	**1708 (-19%)**	1740	
Affine	1658	2030	2072	2043	**1606 (-24%)**	1700	
Nonlin. tensor	2049	2102	2100	2076	**1911 (-9%)**	1957	
Nonlin. MLP-ANN	1673	2059	2089	2059	**1629 (-23%)**	1644	
No-enhancement	2102	2715	2702	2686	2081	1849	

Table 7.6: German data: test MAPE [%] of the proposed aggregation techniques: missing-experts scenario. The best results are highlighted in bold; between parentheses, the percentage decrease with respect to the best not-enhanced expert that uses all data (lower left corner).

Individual Enhancement	**Best expert**	**AEE**					**EAE**
		Simple average	**Trimm. average**	**Winsor. average**	**Constr. Least Squares**	**MLP**	
Additive	2.30	2.83	2.88	2.84	**2.26 (-19%)**	2.30	**2.23 (-20%)**
Multiplicative	2.31	2.88	2.92	2.89	**2.28 (-19%)**	2.35	
Affine	2.18	2.60	2.62	2.60	**2.12 (-24%)**	2.20	
Nonlin. tensor	2.45	2.56	2.57	2.55	**2.27 (-19%)**	2.46	
Nonlin. MLP-ANN	2.22	2.65	2.66	2.64	**2.13 (-24%)**	2.16	
No-enhancement (full information)	2.80	3.91	3.87	3.85	2.95	2.62	

adopted to deal with these phenomena without need to re-calibrate the aggregation models [108], [109]. The conditional mean imputation method uses the first and second order moments of the joint distribution of the experts, in order to derive linear imputation formulas predicting one or more experts from the knowledge of the others.

In order to test the robustness of the proposed aggregation strategies, in the test data 40% of the expert predictions were randomly labelled as missing. The resulting MAPEs are reported in Table 7.6. Again, the best results are obtained by the Affine-CLS AEE. The AEE schemes achieve a MAPE ranging from 2.12% to 2.92%, while the EAE scheme achieves a MAPE = 2.23%. As expected, the MAPE and the RMSE of both the AEE and the EAE increase with respect to the full information scenario. Nevertheless, both the AEE and the EAE, even when applied to the missing data scenario, are still able to improve over the best expert that uses all data, which demonstrates the remarkable robustness of the two enhanced aggregation strategies. The overall robustness of the AEE and the EAE is confirmed also when the RMSEs are considered, see Table 7.7.

Table 7.7: German data: test RMSE [MW] of the proposed aggregation techniques: missing-experts scenario. The best results are highlighted in bold; between parentheses, the percentage decrease with respect to the best not-enhanced expert that uses all data (lower left corner).

		AEE					**EAE**
Individual Enhancement	**Best expert**	**Simple average**	**Trimm. average**	**Winsor. average**	**Constr. Least Squares**	**MLP**	
Additive	1876	2229	2273	2244	**1844 (-12%)**	1900	**1897 (-10%)**
Multiplicative	1873	2241	2288	2256	**1840 (-12%)**	1922	
Affine	1784	2072	2108	2080	**1732 (-18%)**	1800	
Nonlin. tensor	2248	2118	2123	2103	**1978 (-6%)**	2103	
Nonlin. MLP-ANN	1797	2104	2122	2099	**1742 (-17%)**	1798	
No-enhancement (full information)	2102	2728	2722	2702	2192	2054	

Enhancement vs no-enhancement

A major question is whether the forecasting benefits are worth the effort of designing an enhanced aggregation scheme. In order to answer it, the last line of Tables 7.4-7.7 reports the MAPE and RMSE that are obtained if, without performing any enhancement, the experts are just aggregated according to standard methods. The improvement of the AEE and the EAE over the No-Enhancement Aggregation (NEA) is very clear. For example, in the full information case the MAPE of the best NEA scheme (MLP) is 2.38% compared to 1.95% and 1.98% achieved by the Affine-CLS AEE and the EAE, respectively. Not only the best AEE and EAE schemes outperform NEA, but even most of the non optimal AEE end EAE schemes provide a definite improvement. This superiority of enhanced schemes vs not-enhanced ones holds for both the full information and the missing-experts scenarios.

7.4 Discussion

In the time series forecasting literature, it is known that aggregating forecasts (herein called 'experts') coming from different external providers can significantly improve the final accuracy. However, possible drifts of the joint distribution of targets and features, associated with the difficulty of retuning the experts' models, can cause a drastical drop in the performances of the aggregated prediction.

In the chapter, we assume the availability of some experts that may need recalibration, but cannot be retrained, possibly because the complete raw data are no more available or the predictors are used as black-boxes, without having access to their inner structure. In addition to the experts, some (typically few) exogenous features are available to the user. The challenge is to combine the primary data (the features) with the secondary ones (the experts) in order to obtain the best possible aggregation.

In this chapter two general nonlinear approaches to expert aggregation are proposed. The main novelty with respect to the existing literature is the introduction of some form of nonlinear 'enhancement', i.e. the exploitation of the exogenous features in order to fix errors and biases of the single experts. Two strategies are proposed and implemented. The Aggregation of Enhanced Experts (AEE) is a two-stage method where the single experts are individually enhanced before being aggregated. The second strategy, Enhanced Aggregation of Experts (EAE) performs a unique nonlinear enhancement of the experts' predictions.

On the German load problem, the AEE and the EAE achieved comparable results, achieving a $\sim 30\%$ reduction of the MAPE and $\sim 24\%$ reduction of the RMSE, compared to the performance of the best expert. In such a case, the AEE could be preferable, because, especially if a linear aggregation is used, the prediction mechanism is much more transparent than with the EAE. In case of large prediction errors, for instance, it is straightforward to isolate the failing experts, which could help the search for the root causes of the poor performance.

A further motivation for the development of smart aggregation techniques is the case of missing experts. Again with reference to the German load benchmark, a scenario where 40% of the experts are randomly missing was considered. It is remarkable that both the AEE and the EAE prove robust in this rather extreme case. In fact, the AEE and the EAE applied to the missing-experts scenario still guarantee a $19-24\%$ reduction of the MAPE and a $6-18\%$ reduction of the RMSE with respect to the best expert in the full information scenario.

The German load benchmark shows that the introduction of enhancement schemes can bring a definite advantage with respect to the simple aggregation of the experts according to standard methods. For instance, in the full information case the MAPE goes from 2.38% for the MLP aggregation without enhancement to 1.95% with the AEE, corresponding to a neat 18% improvement.

In conclusion, smart aggregation strategies leveraging on some form of enhancement appear to be not only generally advantageous but also robust with respect to missing-experts scenarios.

8

Conclusions

Electrical load forecasting remains a crucial task in nowadays distribution grids because of its role in the stability and security of the system and its financial use within the energy market. In this thesis, both long and short-term forecasting were addressed, resorting to an integrated approach that implements diverse techniques from statistics, Fourier analysis and machine learning.

Concerning long-term forecasting, spectral analyis was interweaved with machine learning to obtain an efficient LASSO-FFT algorithm that, with $O\left(n \log(n)\right)$ complexity, yields a parsimonious frequency domain description of the multi-seasonal component of the quarter-hourly Italian electric load. The results demonstrate, on one hand, that the yearly and weekly patterns, rather than having just an additive effect, interact with each other. Second and more important, we found that the multi-seasonal component is remarkably stable through the years and accounts for a significant fraction of the load, which makes it possible to produce rather good predictions weeks and months ahead.

Short-term forecasting was investigated both for normal days and a subclass of special days, i.e. the Easter week ones, still using the Italian demand data as case study. In the former case, smoothing techniques such as Tikhonov regularization and Radial Basis Functions were applied to reduce the variance of a linear multipredictor scheme. For the Easter Week demand, a Gaussian Process method that uses a statistically estimated kernel was used to track the departure of the load demand from an average one. The main finding is that, by properly exploiting the serial correlation of the load demand series, accurate short-term predictions are indeed possible, even without including in the model structure exogenous factors such as weather forecasts.

Finally, the potential of aggregating forecasts has been explored in this thesis. In particular even a simple average aggregation of uncorrelated forecasts for the short-term prediction of the Italian load lead to a substantial improvement within the performance indexes for both normal days and Easter Week. This topic was further explored in the last chapter where more sophisticated aggregation strategies were designed in order to combine the predictions of different experts. The main finding is that smart aggregation strategies offer a powerful means to produce accurate and robust predictions.

Appendix A

Italian special days

The set $\mathcal{D}_s$ of special days includes Winter, Summer, Easter and national holidays and their associated windows of influence. The days included in this set are summarized below.

Winter holidays

Winter holidays account for Christmas Eve, Christmas, St. Stephen's Day, New Year's Eve, New Year and Epiphany holidays and include all days within December 22 and January 6, for a total of 16 days.

Summer holidays

Summer holidays consists of a range of three weeks around August 15, in particular from August 5 to August 24 (20 days).

National holidays

The national holidays in Italy are: Liberation Day (April 25), Labour Day (May 1), Republic Day (June 2), All Saints' Day (November 1), Feast of the Immaculate Conception (December 8). For each national holiday, a window of influence of five days is considered (two days before and after the holiday itself), for a total of 25 days.

Easter holidays

Easter represent a particular case of holiday since its date is not fixed but varies within March and April, while its weekday is always Sunday. In this thesis, a window of 5 days is associated to

Easter holidays, in particular from the Thursday before to the Monday after (Easter Monday). According to this convention, the following table summarizes the dates of the Easter holidays of the years 1990-2019.

Table A.1: Easter holiday dates: years 1990-2019.

Year	**Date**
1990	April 12 - April 16
1991	March 28 - April 1
1992	April 16 - April 20
1993	April 8 - April 12
1994	March 31 - April 4
1995	April 13 - April 17
1996	April 4 - April 8
1997	March 27 - March 31
1998	April 9 - April 13
1999	April 1 - April 5
2000	April 20 - April 24
2001	April 12 - April 16
2002	March 28 - April 1
2003	April 17 - April 21
2004	April 8 - April 12
2005	March 24 - March 28
2006	April 13 - April 17
2007	April 5 - April 9
2008	March 20 - March 24
2009	April 9 - April 13
2010	April 1 - April 5
2011	April 21 - April 25
2012	April 5 - April 9
2013	March 28 - April 1
2014	April 17 - April 21
2015	April 2 - April 6
2016	March 24 - March 28
2017	April 13 - April 17
2018	March 29 - April 2
2019	April 18 - April 22

Appendix B

Italian electric load short-term prediction: performances in yearly seasons and daily phases

The test performances of the proposed Italian electric load short-term predictors on normal days, shown in the tables in Section 5.5, highlight the tendency to improve mainly the quarter-hourly performance indexes rather than the daily ones. Moreover, among the considered indexes, it can be seen that typically the $RMSE$ has a lower improvement (and in the 2017 scenario it even gets worse) with respect to the $MAPE$ and the MAE. This could be a consequence of a few large errors within the proposed forecasts that strongly affect the $RMSE$ (since it goes with the square of the error).

In the following tables, the $MAPE$, $RMSE$ and MAE are computed in the following two scenarios: (i) for the four seasons of the year (Spring: March, April, May; Summer: June, July, August; Fall: September, October, November; Winter: December, January, February); (ii) for the four phases of day (Morning: from 8:00 to 12:45; Afternoon: from 13:00 to 18:45; Evening: from 19:00 to 01:45; Night: from 02:00 to 7:45). It turns out that the newly proposed methods perform worse during the morning and the afternoon and during Summer and Winter. An explanation is the possible effect of weather variables, such as the temperature that is not considered in any of the proposed models, but is most probably included in the Terna predictor.

Table B.1: Forecasts performances on the Spring of 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\textbf{daily}}$[%]	**RMSE**$_{\textbf{daily}}$[**GW**]	**MAE**$_{\textbf{daily}}$[**GW**]
Terna	2.39	1.14	0.88	1.61	0.76	0.59
OLS	3.07 (28%)	1.45 (0%)	1.1 (25%)	2.32 (44%)	1.02 (0%)	0.81 (37%)
TA	2.19 (−8%)	1.07 (−6%)	0.8 (−9%)	1.7 (6%)	0.8 (5%)	0.61 (3%)
TS	2.2 (−8%)	1.07 (−6%)	0.8 (−9%)	1.72 (7%)	0.8 (5%)	0.62 (5%)
RBF	2.21 (−8%)	1.08 (−5%)	0.81 (−8%)	1.75 (9%)	0.81 (7%)	0.63 (7%)
TE	**2.15 (−10%)**	**1.05 (−8%)**	**0.78 (−11%)**	1.66 (3%)	0.76 (0%)	0.6 (2%)
OnE	2.3 (−4%)	1.09 (−4%)	0.82 (−7%)	**1.57 (−2%)**	**0.75 (−1%)**	**0.57 (−3%)**

Table B.2: Forecasts performances on the Summer of 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\textbf{daily}}$[%]	**RMSE**$_{\textbf{daily}}$[**GW**]	**MAE**$_{\textbf{daily}}$[**GW**]
Terna	2.35	**1.22**	0.97	**1.65**	**0.83**	**0.69**
OLS	2.95 (26%)	1.78 (0%)	1.25 (28%)	2.2 (33%)	1.24 (0%)	0.89 (28%)
TA	2.35 (0%)	1.44 (18%)	0.98 (1%)	1.79 (8%)	0.99 (19%)	0.72 (4%)
TS	**2.34 (0%)**	1.43 (17%)	**0.97 (0%)**	1.76 (7%)	0.98 (18%)	0.71 (3%)
RBF	2.34 (0%)	1.44 (18%)	0.97 (0%)	1.77 (7%)	0.98 (18%)	0.72 (4%)
TE	2.62 (11%)	1.55 (27%)	1.09 (12%)	2.23 (35%)	1.11 (34%)	0.89 (28%)
OnE	4.27 (82%)	2.16 (77%)	1.7 (75%)	3.67 (122%)	1.82 (119%)	1.46 (112%)

Table B.3: Forecasts performances on the Fall of 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\textbf{daily}}$[%]	**RMSE**$_{\textbf{daily}}$[**GW**]	**MAE**$_{\textbf{daily}}$[**GW**]
Terna	1.96	0.92	0.71	**0.87**	**0.42**	**0.31**
OLS	2.27 (16%)	1.12 (0%)	0.84 (18%)	1.41 (62%)	0.64 (0%)	0.51 (65%)
TA	**1.71** (**−13**%)	**0.85** (**−8**%)	**0.62** (**−13**%)	1.13 (30%)	0.51 (21%)	0.4 (28%)
TS	1.73 (−12%)	0.86 (−7%)	0.63 (−11%)	1.11 (28%)	0.51 (21%)	0.39 (26%)
RBF	1.74 (−11%)	0.87 (−5%)	0.63 (−11%)	1.13 (30%)	0.51 (21%)	0.4 (28%)
TE	1.79 (−9%)	0.92 (0%)	0.65 (−8%)	1.22 (40%)	0.59 (40%)	0.43 (39%)
OnE	2.19 (12%)	1.11 (21%)	0.77 (8%)	1.59 (83%)	0.85 (102%)	0.55 (77%)

Table B.4: Forecasts performances on the Winter of 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE**$_{\textbf{daily}}$[%]	**RMSE**$_{\textbf{daily}}$[**GW**]	**MAE**$_{\textbf{daily}}$[**GW**]
Terna	2.03	1.0	0.78	**0.9**	**0.43**	**0.35**
OLS	2.4 (18%)	1.3 (0%)	0.94 (21%)	1.71 (90%)	0.88 (100%)	0.65 (86%)
TA	1.72 (−15%)	0.88 (−12%)	**0.66** (**−15**%)	1.21 (34%)	0.6 (40%)	0.46 (31%)
TS	1.71 (−16%)	0.88 (−12%)	0.66 (−15%)	1.2 (33%)	0.6 (40%)	0.45 (28%)
RBF	**1.7** (**−16**%)	**0.87** (**−13**%)	0.66 (−15%)	1.19 (32%)	0.59 (37%)	0.45 (28%)
TE	1.75 (−14%)	0.92 (−8%)	0.68 (−13%)	1.27 (41%)	0.65 (51%)	0.48 (37%)
OnE	1.84 (−9%)	0.89 (−11%)	0.7 (−10%)	1.36 (51%)	0.62 (44%)	0.52 (49%)

TABLE B.5: Forecasts performances in the Morning on 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]
Terna	2.4	1.26	1.03
OLS	2.94 (23%)	1.64 (0%)	1.25 (21%)
TA	2.14 (−11%)	1.17 (−7%)	**0.89** (−**14**%)
TS	**2.12** (−**12**%)	**1.16** (−**8**%)	0.89 (−14%)
RBF	2.13 (−11%)	1.17 (−7%)	0.89 (−14%)
TE	2.24 (−7%)	1.21 (−4%)	0.94 (−9%)
OnE	2.72 (13%)	1.54 (22%)	1.14 (11%)

TABLE B.6: Forecasts performances in the Afternoon on 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]
Terna	**2.1**	**1.14**	**0.9**
OLS	3.13 (49%)	1.75 (0%)	1.33 (48%)
TA	2.32 (10%)	1.32 (16%)	0.98 (9%)
TS	2.33 (11%)	1.32 (16%)	0.98 (9%)
RBF	2.32 (10%)	1.32 (16%)	0.98 (9%)
TE	2.43 (16%)	1.39 (22%)	1.02 (13%)
OnE	2.73 (30%)	1.63 (43%)	1.15 (28%)

TABLE B.7: Forecasts performances in the Evening on 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]
Terna	**1.86**	**0.84**	**0.65**
OLS	2.53 (36%)	1.28 (0%)	0.91 (40%)
TA	1.91 (3%)	1.06 (26%)	0.69 (6%)
TS	1.93 (4%)	1.07 (27%)	0.7 (8%)
RBF	1.93 (4%)	1.07 (27%)	0.7 (8%)
TE	2.05 (10%)	1.15 (37%)	0.75 (15%)
OnE	2.59 (39%)	1.32 (56%)	0.92 (42%)

TABLE B.8: Forecasts performances at Night on 2017, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]
Terna	2.27	0.97	0.72
OLS	1.92 (−15%)	0.76 (0%)	0.58 (−19%)
TA	**1.5** (**−34**%)	0.61 (−37%)	**0.45** (**−37**%)
TS	1.5 (−34%)	0.61 (−37%)	0.45 (−37%)
RBF	1.51 (−33%)	0.61 (−37%)	0.45 (−37%)
TE	1.51 (−33%)	**0.6** (**−38**%)	0.45 (−37%)
OnE	2.51 (11%)	1.03 (6%)	0.74 (3%)

Table B.9: Forecasts performances on the Spring of 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[**GW**]	$\mathbf{MAE_{daily}}$[**GW**]
Terna	**2.19**	**1.05**	**0.8**	**1.25**	**0.59**	**0.46**
OLS	3.39 (55%)	1.76 (0%)	1.28 (60%)	2.69 (114%)	1.23 (100%)	0.99 (114%)
TA	2.51 (15%)	1.39 (32%)	0.94 (17%)	1.97 (57%)	0.97 (64%)	0.71 (54%)
TS	2.52 (15%)	1.41 (34%)	0.95 (19%)	1.99 (59%)	0.98 (66%)	0.72 (56%)
RBF	2.49 (14%)	1.39 (32%)	0.93 (16%)	1.95 (56%)	0.97 (64%)	0.7 (52%)
TE	2.32 (6%)	1.23 (17%)	0.86 (7%)	1.68 (34%)	0.79 (34%)	0.59 (28%)
OnE	2.74 (25%)	1.43 (36%)	1.02 (27%)	2.07 (66%)	0.93 (57%)	0.75 (63%)

Table B.10: Forecasts performances on the Summer of 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[**GW**]	$\mathbf{MAE_{daily}}$[**GW**]
Terna	2.4	1.19	0.93	1.83	0.9	0.72
OLS	2.76 (15%)	1.5 (0%)	1.13 (22%)	2.03 (11%)	1.02 (0%)	0.81 (13%)
TA	1.96 (−18%)	1.08 (−9%)	**0.8** (**−14**%)	1.53 (−16%)	0.74 (−18%)	**0.6** (**−17**%)
TS	1.96 (−18%)	1.07 (−10%)	0.8 (−14%)	**1.52** (**−17**%)	**0.72** (**−20**%)	0.6 (−17%)
RBF	**1.95** (**−19**%)	**1.06** (**−11**%)	0.8 (−14%)	1.52 (−17%)	0.72 (−20%)	0.6 (−17%)
TE	2.06 (−14%)	1.14 (−4%)	0.84 (−10%)	1.67 (−9%)	0.8 (−11%)	0.66 (−8%)
OnE	2.93 (22%)	1.5 (26%)	1.17 (26%)	2.44 (33%)	1.19 (32%)	0.97 (35%)

Table B.11: Forecasts performances on the Fall of 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE$_{\text{daily}}$**[%]	**RMSE$_{\text{daily}}$**[**GW**]	**MAE$_{\text{daily}}$**[**GW**]
Terna	2.51	1.12	0.91	1.85	0.77	0.67
OLS	2.36 (−6%)	1.21 (0%)	0.9 (−1%)	1.58 (−15%)	0.76 (0%)	0.59 (−12%)
TA	1.58 (−37%)	0.82 (−27%)	0.6 (−34%)	1.21 (−35%)	0.58 (−25%)	0.45 (−33%)
TS	1.6 (−36%)	0.83 (−26%)	0.61 (−33%)	1.25 (−32%)	0.59 (−23%)	0.46 (−31%)
RBF	1.58 (−37%)	0.82 (−27%)	0.6 (−34%)	1.19 (−36%)	0.57 (−26%)	0.44 (−34%)
TE	**1.54** (**−39**%)	**0.79** (**−28**%)	**0.58** (**−36**%)	**1.18** (**−36**%)	**0.54** (**−30**%)	**0.43** (**−36**%)
OnE	1.77 (−28%)	0.91 (−19%)	0.66 (−27%)	1.3 (−30%)	0.67 (−13%)	0.48 (−28%)

Table B.12: Forecasts performances on the Winter of 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]	**MAPE$_{\text{daily}}$**[%]	**RMSE$_{\text{daily}}$**[**GW**]	**MAE$_{\text{daily}}$**[**GW**]
Terna	2.49	1.2	0.95	1.57	0.71	0.6
OLS	2.6 (4%)	1.37 (0%)	1.02 (7%)	1.89 (20%)	0.96 (0%)	0.72 (20%)
TA	1.79 (−28%)	0.97 (−19%)	0.7 (−26%)	1.3 (−17%)	0.68 (−4%)	0.49 (−18%)
TS	1.77 (−28%)	0.96 (−20%)	0.69 (−27%)	1.26 (−20%)	0.66 (−7%)	0.48 (−20%)
RBF	1.76 (−28%)	0.96 (−20%)	0.69 (−27%)	1.26 (−20%)	0.66 (−7%)	0.47 (−22%)
TE	**1.67** (**−33**%)	**0.9** (**−25**%)	**0.65** (**−32**%)	**1.15** (**−27**%)	**0.59** (**−17**%)	**0.43** (**−28**%)
OnE	2.01 (−19%)	1.05 (−12%)	0.78 (−18%)	1.41 (−10%)	0.73 (3%)	0.54 (−10%)

Table B.13: Forecasts performances in the Morning on 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]
Terna	2.52	1.36	1.07
OLS	3.0 (19%)	1.66 (0%)	1.28 (20%)
TA	2.22 (−12%)	1.28 (−6%)	0.94 (−12%)
TS	2.24 (−11%)	1.29 (−5%)	0.94 (−12%)
RBF	2.21 (−12%)	1.27 (−7%)	0.93 (−13%)
TE	**2.18** (**−13%**)	**1.21** (**−11%**)	**0.92** (**−14%**)
OnE	2.57 (2%)	1.46 (7%)	1.1 (3%)

Table B.14: Forecasts performances in the Afternoon on 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]
Terna	**2.08**	**1.13**	**0.88**
OLS	3.46 (66%)	1.89 (0%)	1.48 (68%)
TA	2.43 (17%)	1.38 (22%)	1.03 (17%)
TS	2.44 (17%)	1.38 (22%)	1.03 (17%)
RBF	2.41 (16%)	1.37 (21%)	1.02 (16%)
TE	2.34 (12%)	1.3 (15%)	0.99 (12%)
OnE	2.71 (30%)	1.56 (38%)	1.16 (32%)

Table B.15: Forecasts performances in the Evening on 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]
Terna	2.96	1.2	1.02
OLS	2.6 (−12%)	1.26 (0%)	0.94 (−8%)
TA	1.74 (−41%)	0.88 (−27%)	0.63 (−38%)
TS	1.73 (−42%)	0.87 (−27%)	0.63 (−38%)
RBF	1.7 (−43%)	0.86 (−28%)	0.61 (−40%)
TE	**1.62** (−**45**%)	**0.81** (−**32**%)	**0.58** (−**43**%)
OnE	1.99 (−33%)	0.92 (−23%)	0.7 (−31%)

Table B.16: Forecasts performances at Night on 2018, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]
Terna	2.08	0.79	0.63
OLS	1.92 (−8%)	0.73 (0%)	0.57 (−10%)
TA	**1.32** (−**37**%)	**0.51** (−**35**%)	**0.39** (−**38**%)
TS	1.32 (−37%)	0.51 (−35%)	0.39 (−38%)
RBF	1.34 (−36%)	0.52 (−34%)	0.4 (−37%)
TE	1.34 (−36%)	0.53 (−33%)	0.4 (−37%)
OnE	1.99 (−4%)	0.76 (−4%)	0.59 (−6%)

Table B.17: Forecasts performances on the Spring of 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	**MAPE$_{\textbf{daily}}$**[%]	**RMSE$_{\textbf{daily}}$**[GW]	**MAE$_{\textbf{daily}}$**[GW]
Terna	2.67	1.16	0.95	2.09	0.83	0.74
OLS	2.78 (4%)	1.44 (0%)	1.02 (7%)	2.18 (4%)	1.0 (0%)	0.78 (5%)
TA	**1.78** (**−33**%)	**0.87** (**−25**%)	**0.64** (**−33**%)	**1.19** (**−43**%)	**0.54** (**−35**%)	**0.43** (**−42**%)
TS	1.82 (−32%)	0.91 (−22%)	0.66 (−31%)	1.28 (−39%)	0.59 (−28%)	0.46 (−38%)
RBF	1.81 (−32%)	0.91 (−22%)	0.66 (−31%)	1.26 (−40%)	0.59 (−28%)	0.45 (−39%)
TE	1.84 (−31%)	0.92 (−21%)	0.67 (−28%)	1.27 (−39%)	0.59 (−28%)	0.45 (−39%)
OnE	2.08 (−22%)	1.01 (−13%)	0.75 (−21%)	1.4 (−33%)	0.62 (−25%)	0.51 (−31%)

Table B.18: Forecasts performances on the Summer of 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	**MAPE$_{\textbf{daily}}$**[%]	**RMSE$_{\textbf{daily}}$**[GW]	**MAE$_{\textbf{daily}}$**[GW]
Terna	2.68	1.39	1.07	2.12	1.04	0.86
OLS	2.77 (3%)	1.53 (0%)	1.16 (8%)	2.31 (9%)	1.09 (0%)	0.93 (8%)
TA	2.46 (−8%)	1.42 (2%)	1.03 (−4%)	2.12 (0%)	1.06 (2%)	0.86 (0%)
TS	**2.31** (**−14**%)	**1.32** (**−5**%)	**0.96** (**−10**%)	**1.91** (**−10**%)	**0.95** (**−9**%)	**0.77** (**−10**%)
RBF	2.31 (−14%)	1.33 (−4%)	0.96 (−10%)	1.91 (−10%)	0.96 (−8%)	0.77 (−10%)
TE	2.46 (−8%)	1.43 (3%)	1.03 (−4%)	2.14 (1%)	1.06 (2%)	0.86 (0%)
OnE	3.52 (31%)	1.84 (32%)	1.44 (35%)	3.09 (46%)	1.54 (48%)	1.27 (48%)

Table B.19: Forecasts performances on the Fall of 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[GW]	$\mathbf{MAE_{daily}}$[GW]
Terna	2.35	1.08	0.85	1.6	0.73	0.58
OLS	2.36 (0%)	1.17 (0%)	0.87 (2%)	1.84 (15%)	0.81 (0%)	0.65 (12%)
TA	1.66 (−28%)	0.82 (−24%)	0.61 (−28%)	1.12 (−30%)	0.52 (−28%)	0.41 (−28%)
TS	**1.59** (**−32**%)	**0.8** (**−26**%)	**0.59** (**−31**%)	1.05 (−34%)	**0.49** (**−33**%)	0.38 (−34%)
RBF	1.6 (−32%)	0.8 (−26%)	0.59 (−31%)	**1.04** (**−35**%)	0.49 (−33%)	**0.37** (**−36**%)
TE	1.74 (−26%)	0.88 (−19%)	0.65 (−24%)	1.32 (−18%)	0.59 (−19%)	0.48 (−17%)
OnE	2.1 (−11%)	0.99 (−8%)	0.77 (−9%)	1.54 (−4%)	0.7 (−4%)	0.56 (−3%)

Table B.20: Forecasts performances on the Winter of 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]	$\mathbf{MAPE_{daily}}$[%]	$\mathbf{RMSE_{daily}}$[GW]	$\mathbf{MAE_{daily}}$[GW]
Terna	2.48	1.17	0.94	1.87	0.81	0.71
OLS	2.2 (−11%)	1.18 (0%)	0.88 (−6%)	1.66 (−11%)	0.8 (0%)	0.64 (−10%)
TA	1.76 (−28%)	0.92 (−21%)	0.69 (−27%)	1.37 (−27%)	0.67 (−17%)	0.52 (−27%)
TS	1.72 (−31%)	0.91 (−22%)	0.67 (−28%)	1.34 (−28%)	0.66 (−19%)	0.51 (−28%)
RBF	1.72 (−31%)	0.91 (−22%)	0.67 (−28%)	1.34 (−28%)	0.66 (−19%)	0.51 (−28%)
TE	**1.66** (**−33**%)	**0.89** (**−24**%)	**0.65** (**−31**%)	**1.24** (**−34**%)	**0.62** (**−23**%)	**0.47** (**−34**%)
OnE	2.0 (−19%)	1.04 (−11%)	0.77 (−18%)	1.57 (−16%)	0.78 (−4%)	0.6 (−15%)

Table B.21: Forecasts performances in the Morning on 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]
Terna	2.54	1.35	1.07
OLS	2.82 (11%)	1.5 (0%)	1.19 (11%)
TA	**2.02** (**−20**%)	**1.1** (**−19**%)	**0.86** (**−20**%)
TS	2.02 (−20%)	1.1 (−19%)	0.86 (−20%)
RBF	2.02 (−20%)	1.1 (−19%)	0.86 (−20%)
TE	2.11 (−17%)	1.16 (−14%)	0.9 (−16%)
OnE	2.63 (4%)	1.45 (7%)	1.13 (6%)

Table B.22: Forecasts performances in the Afternoon on 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[GW]	**MAE**[GW]
Terna	**2.21**	**1.21**	**0.94**
OLS	3.27 (48%)	1.75 (0%)	1.38 (47%)
TA	2.3 (4%)	1.29 (7%)	0.97 (3%)
TS	2.27 (3%)	1.26 (4%)	0.96 (2%)
RBF	2.27 (3%)	1.26 (4%)	0.96 (2%)
TE	2.41 (9%)	1.33 (10%)	1.02 (9%)
OnE	2.6 (18%)	1.51 (25%)	1.12 (19%)

Table B.23: Forecasts performances in the Evening on 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]
Terna	3.27	1.34	1.13
OLS	2.32 (−28%)	1.14 (0%)	0.84 (−26%)
TA	1.9 (−42%)	1.01 (−25%)	0.69 (−39%)
TS	**1.74** (**−47**%)	**0.92** (**−31**%)	**0.63** (**−44**%)
RBF	1.75 (−46%)	0.93 (−31%)	0.63 (−44%)
TE	1.82 (−44%)	0.98 (−27%)	0.66 (−42%)
OnE	2.25 (−31%)	1.09 (−19%)	0.79 (−30%)

Table B.24: Forecasts performances at Night on 2019, with the percentage variation with respect to Terna results between brackets.

	MAPE[%]	**RMSE**[**GW**]	**MAE**[**GW**]
Terna	2.09	0.8	0.63
OLS	1.65 (−21%)	0.65 (0%)	0.49 (−22%)
TA	1.36 (−35%)	0.53 (−34%)	0.41 (−35%)
TS	**1.31** (**−37**%)	**0.52** (**−35**%)	**0.39** (**−38**%)
RBF	1.31 (−37%)	0.52 (−35%)	0.39 (−38%)
TE	1.31 (−37%)	0.52 (−35%)	0.39 (−38%)
OnE	2.15 (3%)	0.84 (5%)	0.64 (2%)

Bibliography

[1] M. Jaradat, M. Jarrah, A. Bousselham, Y. Jararweh, and M. Al-Ayyoub, "The internet of energy: Smart sensor networks and big data management for smart grid," vol. 56, 08 2015.

[2] G. I. Webb and Z. Zheng, "Multistrategy ensemble learning: reducing error by combining ensemble learning techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, pp. 980–991, Aug 2004.

[3] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, vol. 14. 01 2012.

[4] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[5] G. Louppe and P. Geurts, "Ensembles on random patches," in *Machine Learning and Knowledge Discovery in Databases* (P. A. Flach, T. De Bie, and N. Cristianini, eds.), (Berlin, Heidelberg), pp. 346–361, Springer Berlin Heidelberg, 2012.

[6] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[7] J. Sill, G. Takacs, L. Mackey, and D. Lin, "Feature-weighted linear stacking," 911.

[8] L. Wang, S. Mao, B. M. Wilamowski, and R. M. Nelms, "Ensemble learning for load forecasting," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 616–628, 2020.

[9] J. Nowotarski, B. Liu, R. Weron, and T. Hong, "Improving short term load forecast accuracy via combining sister forecasts," *Energy*, vol. 98, pp. 40–49, 03 2016.

[10] J. B. Predd, D. N. Osherson, S. R. Kulkarni, and H. V. Poor, "Aggregating Probabilistic Forecasts from Incoherent and Abstaining Experts," *Decision Analysis*, vol. 5, pp. 177–189, December 2008.

[11] B. Turner, M. Steyvers, E. Merkle, D. Budescu, and T. Wallsten, "Forecast aggregation via recalibration," *Machine Learning*, 06 2013.

[12] P. Gaillard, Y. Goude, and R. Nedellec, “Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 1038–1050, 2016.

[13] R. Cooke, “The aggregation of expert judgment: Do good things come to those who weight?,” *Risk Analysis*, vol. 35, 02 2015.

[14] K. F. Wallis, “Combining forecasts – forty years later,” *Applied Financial Economics*, vol. 21, no. 1-2, pp. 33–41, 2011.

[15] R. Ranjan and T. Gneiting, “Combining probability forecasts,” *Journal of the Royal Statistical Society Series B*, vol. 72, pp. 71–91, 01 2010.

[16] M. Devaine, P. Gaillard, Y. Goude, and G. Stoltz, “Forecasting electricity consumption by aggregating specialized experts,” 01 2013.

[17] C. Magazzino, “Electricity demand, gdp and employment: evidence from italy,” *Frontiers in Energy*, Mar 2014.

[18] S. Ramos, J. Soares, and Z. Vale, “Short-term load forecasting based on load profiling,” pp. 1–5, 01 2013.

[19] G. A. Mbamalu and M. E. El-Hawary, “Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation,” *IEEE Transactions on Power Systems (Institute of Electrical and Electronics Engineers); (United States)*, 2 1993.

[20] J.-F. Chen, W. ming Wang, and C.-M. Huang, “Analysis of an adaptive time-series autoregressive moving-average (arma) model for short-term load forecasting,” 1995.

[21] S. R. Huang, “Short-term load forecasting using threshold autoregressive models,” *IEE Proceedings - Generation, Transmission and Distribution*, vol. 144, no. 5, pp. 477–481, 1997.

[22] L. J. Soares and M. C. Medeiros, “Modelling and forecasting short-term electricity load: a two step methodology,” Textos para discussão 495, Department of Economics PUC-Rio (Brazil), Feb. 2005.

[23] Hong-Tzer Yang, Chao-Ming Huang, and Ching-Lien Huang, “Identification of armax model for short term load forecasting: an evolutionary programming approach,” in *Proceedings of Power Industry Computer Applications Conference*, pp. 325–330, 1995.

[24] W. Charytoniuk, M. S. Chen, and P. Van Olinda, “Nonparametric regression based short-term load forecasting,” *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 725–730, 1998.

[25] S. Fan and R. J. Hyndman, “Short-term load forecasting based on a semi-parametric additive model,” *IEEE Transactions on Power Systems*, vol. 27, Feb 2012.

[26] V. Dordonnat, A. Pichavant, and A. Pierrot, “GEFCom2014 probabilistic electric load forecasting using time series and semi-parametric regression models,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 1005–1011, 2016.

[27] I. Moghram and S. Rahman, “Analysis and evaluation of five short-term load forecasting techniques,” *IEEE Transactions on Power Systems*, vol. 4, no. 4, pp. 1484–1491, 1989.

[28] H. Al-Hamadi and S. Soliman, “Short-term electric load forecasting based on kalman filtering algorithm with moving window weather and load model,” *Electric Power Systems Research - ELEC POWER SYST RES*, vol. 68, pp. 47–59, 01 2004.

[29] H. Takeda, Y. Tamura, and S. Sato, “Using the ensemble kalman filter for electricity load forecasting and analysis,” *Energy*, vol. 104, pp. 184–198, 06 2016.

[30] P.-H. Kuo and C.-J. Huang, “A high precision artificial neural networks model for short-term energy load forecasting,” *Energies*, vol. 11, p. 213, 01 2018.

[31] S. Ryu, J. Noh, and H. Kim, “Deep neural network based demand side short term load forecasting,” *Energies*, vol. 10, p. 3, 12 2016.

[32] W. Kong, Z. Dong, Y. Jia, D. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on lstm recurrent neural network,” *IEEE Transactions on Smart Grid*, vol. PP, pp. 1–1, 09 2017.

[33] S. Hassan, A. Khosravi, J. Jaafar, and M. Khanesar, “A systematic design of interval type-2 fuzzy logic system using extreme learning machine for electricity load demand forecasting,” *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 1–10, Nov. 2016.

[34] X. Zhang, J. Wang, and K. Zhang, “Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by cuckoo search algorithm,” *Electric Power Systems Research*, vol. 146, pp. 270–285, 05 2017.

[35] H. Jiang, Y. Zhang, E. Muljadi, J. Zhang, and W. Gao, “A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization,” *IEEE Transactions on Smart Grid*, vol. PP, pp. 1949–3053, 11 2016.

[36] D. Srinivasan and M. A. Lee, “Survey of hybrid fuzzy neural approaches to electric load forecasting,” in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 5, Oct 1995.

[37] S. R. Khuntia, J. L. Rueda, and M. A. van der Meijden, "Forecasting the load of electrical power systems in mid- and long-term horizons: a review," *IET Generation, Transmission & Distribution*, vol. 10, pp. 3971–3977(6), December 2016.

[38] L.Ghods and M. Kalantar, "Different methods of long-term electric load demand forecasting; a comprehensive review," *Iranian Journal of Electrical & Electronic Engineering*, vol. 7, December 2011.

[39] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[40] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International Journal of Forecasting*, 2006.

[41] F. Martínez-Álvarez, A. Troncoso, G. Asencio-Cortés, and J. C. Riquelme, "A survey on data mining techniques applied to electricity-related time series forecasting," 2015.

[42] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making.," *European Journal of Operational Research*, 2009.

[43] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Systems Research*, January 2016.

[44] A. Singh, K. Ibraheem, and M. Muazzam, "An overview of electricity demand forecasting techniques," *Proceedings Of-National Conference on Emerging Trends in Electrical, Instrumentation & Communication Engineering*, vol. 3, pp. 38–48, 01 2013.

[45] L. Ljung, *System identification: theory for the user.* Englewood Cliffs, NJ, USA: Prentice-Hall, 1987.

[46] H. Mori, Y. Sone, D. Moridera, and T. Kondo, "Fuzzy inference models for short-term load forecasting with tabu search," vol. 6, pp. 551–556 vol.6, 1999.

[47] T. Senjyu, H. Takara, K. Asato, K. Uezato, and T. F. Funabashi, "Next day load curve forecasting using hybrid correction method," vol. 3, pp. 1701–1706 vol.3, 2002.

[48] S. H. Ling, F. H. F. Leung, H. K. Lam, and P. K. S. Tam, "Short-term electric load forecasting based on a neural fuzzy network," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 6, pp. 1305–1316, 2003.

[49] R. Hyndman and A. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, pp. 679–688, 02 2006.

[50] A. Syntetos and J. Boylan, "The accuracy of intermittent demand estimates," *International Journal of Forecasting*, vol. 21, pp. 303–314, 04 2005.

[51] S. Makridakis and M. Hibon, "The m3-competition: Results, conclusions and implications," *International Journal of Forecasting*, vol. 16, pp. 451–476, 10 2000.

[52] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 8, pp. 69–80, 1992.

[53] R. Fildes, "The evaluation of extrapolative forecasting methods," *International Journal of Forecasting*, vol. 8, pp. 81–98, 06 1992.

[54] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*, pp. 63–71, Springer, 2004.

[55] M. Gruber, *Improving efficiency by shrinkage: The James-Stein and ridge regression estimators.* 01 2017.

[56] A. Saleh, M. Arashi, and B. M. G. Kibria, "Theory of ridge regression estimation with applications," 02 2019.

[57] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, 1994.

[58] G. Wahba, *Spline Models for Observational Data.* Philadelphia: Society for Industrial and Applied Mathematics, 1990.

[59] H. Prautzsch, W. Boehm, and M. Paluszny, *Bezier and B-Spline Techniques.* Berlin, Heidelberg: Springer-Verlag, 2002.

[60] P. H. C. Eilers and B. D. Marx, "Flexible smoothing with b-splines and penalties," *STATISTICAL SCIENCE*, vol. 11, pp. 89–121, 1996.

[61] C. d. Boor, *A Practical Guide to Splines.* New York: Springer Verlag, 1978.

[62] P. Lusis, K. Khalilpour, L. L. H. Andrew, and A. Liebman, "Short-term residential load forecasting: Impact of calendar effects and forecast granularity," *Applied Energy*, November 2017.

[63] G. Dudek, "Forecasting time series with multiple seasonal cycles using neural networks with local learning," in *Artificial Intelligence and Soft Computing*, (Berlin, Heidelberg), Springer Berlin Heidelberg, 2013.

[64] R. Hyndman, K. Ord, R. David Snyder, F. Vahid, P. G. Gould, and A. Koehler, "Forecasting time series with multiple seasonal patterns," *European Journal of Operational Research*, November 2008.

[65] A. M. D. Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, 2011.

[66] J. W. Taylor, "Exponentially weighted methods for forecasting intraday time series with multiple seasonal cycles," *International Journal of Forecasting*, vol. 26, 2010.

[67] R. J. Hyndman and S. Fan, "Density forecasting for long-term peak electricity demand," *IEEE Transactions on Power Systems*, vol. 25, May 2010.

[68] H. Al-Hamadi and S. Soliman, "Long-term/mid-term electric load forecasting based on short-term correlation and annual growth," *Electric Power Systems Research*, vol. 74, 2005.

[69] A. Guerini and G. D. Nicolao, "Long-term electric load forecasting: A torus-based approach," in *2015 European Control Conference (ECC)*, July 2015.

[70] A. Guerini and G. D. Nicolao, "Long- and short-term electric load forecasting on quarter-hour data: A 3-torus approach," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, June 2016.

[71] J. Fiot and F. Dinuzzo, "Electricity demand forecasting by multi-task learning," *IEEE Transactions on Smart Grid*, March 2018.

[72] P. Babu and P. Stoica, "Spectral analysis of nonuniformly sampled data - a review," *Digital Signal Processing*, 2010.

[73] P. Stoica, J. Li, and J. Ling, "Missing data recovery via a nonparametric iterative adaptive approach," *IEEE Signal Process. Lett.*, 2009.

[74] J. Nowicka-Zagrajek and R. Weron, "Modeling electricity loads in california: Arma models with hyperbolic noise," *Signal Processing*, vol. 82, 2002.

[75] H. Zou, T. Hastie, and R. Tibshirani, "On the 'degrees of freedom' of the lasso," *Ann. Statist.*, 2007.

[76] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, 2008.

[77] A. Bibi, H. Itani, and B. Ghanem, "Fftlasso: Large-scale lasso in the fourier domain," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[78] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, "Solving structured sparsity regularization with proximal methods," in *Machine Learning and Knowledge Discovery in Databases*, 2010.

[79] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM J. Img. Sci.*

[80] K. Bredies and D. A. Lorenz, “Iterative soft-thresholding converges linearly,” 2007.

[81] Y. E. Nesterov, “A method for solving the convex programming problem with convergence rate $o(1/k^2)$,” *Dokl. Akad. Nauk SSSR*, 1983.

[82] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice.* OTexts, 2014.

[83] M. T. Hagan and S. M. Behr, “The time series approach to short-term load forecasting,” *IEEE Power Engineering Review*, vol. PER-7, no. 8, pp. 56–57, 1987.

[84] R. Sood, I. Koprinska, and V. Agelidis, “Electricity load forecasting based on autocorrelation analysis,” pp. 1 – 8, 08 2010.

[85] V. Yadav and D. Srinivasan, “Autocorrelation based weighing strategy for short-term load forecasting with the self-organizing map,” *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010*, vol. 1, 02 2010.

[86] I. Koprinska, M. Rana, and V. Agelidis, “Correlation and instance based feature selection for electricity load forecasting,” *Knowledge-Based Systems*, vol. 82, 02 2015.

[87] “Terna website.” `https://www.terna.it/it/sistema-elettrico/transparency-report/total-load`.

[88] A. Incremona and G. De Nicolao, “Spectral characterization of the multi-seasonal component of the italian electric load: A lasso-fft approach,” *IEEE Control Systems Letters*, vol. PP, pp. 1–1, 06 2019.

[89] Chen Hong and Liu Jianwei, “A weighted multi-model short-term load forecasting system,” in *POWERCON ’98. 1998 International Conference on Power System Technology. Proceedings (Cat. No.98EX151)*, vol. 1, pp. 557–561 vol.1, 1998.

[90] O. Ahmia and N. Farah, “Multi-model approach for electrical load forecasting,” 11 2015.

[91] D. C. Hoaglin and R. E. Welsch, “The hat matrix in regression and anova,” *The American Statistician*, vol. 32, no. 1, pp. 17–22, 1978.

[92] D. Ranaweera, N. Hubele, and A. Papalexopoulos, “Application of radial basis function neural network model for short-term load forecasting,” *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 142, pp. 45 – 50, 02 1995.

[93] S. Konishi, “Bayesian information criteria and smoothing parameter selection in radial basis function networks,” *Biometrika*, vol. 91, pp. 27–43, 02 2004.

[94] G. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* Holden-Day, 1976.

[95] F. Wallis and Bede, *Bede, the Reckoning of Time (Translated Texts for Historians; V. 29).* Liverpool University Press, 1999.

[96] D. Vu, K. Muttaqi, A. Agalgaonkar, and A. Bouzerdoum, "Short-term electricity demand forecasting using autoregressive based time varying model incorporating representative data adjustment," *Applied Energy*, vol. 205, pp. 790–801, 11 2017.

[97] G. Nicolao, M. Pozzi, E. Soda, and M. Stori, "Short-term load forecasting: A power-regression approach," *2014 International Conference on Probabilistic Methods Applied to Power Systems, PMAPS 2014 - Conference Proceedings*, 11 2014.

[98] G. Chicco, R. Napoli, and F. Piglione, "Load pattern clustering for short-term load forecasting of anomalous days," in *2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502)*, vol. 2, pp. 6 pp. vol.2–, 2001.

[99] F. J. Marin, F. Garcia-Lagos, G. Joya, and F. Sandoval, "Global model for short-term load forecasting using artificial neural networks," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 149, no. 2, pp. 121–125, 2002.

[100] S. Yerpude and T. Singhal, "Impact of internet of things (iot) data on demand forecasting," *Indian Journal of Science and Technology*, vol. 10, 05 2017.

[101] J. Hox and H. Boeije, "Data collection, primary versus secondary.," *Encyclopedia of Social Measurement*, vol. 1, 12 2005.

[102] "Entso-e transparency platform." `https://transparency.entsoe.eu/`.

[103] T. Hastie and R. Tibshirani, "Generalized additive models: Some applications," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 371–386, 1987.

[104] A. Pierrot and Y. Goude, "Short-term electricity load forecasting with generalized additive models," 01 2011.

[105] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[106] V. Genre, G. Kenny, A. Meyler, and A. Timmermann, "Combining expert forecasts: Can anything beat the simple average?," *International Journal of Forecasting*, vol. 29, p. 108–121, 03 2013.

[107] C. Fraley, A. Raftery, and T. Gneiting, "Calibrating multimodel forecast ensembles with exchangeable and missing members using bayesian model averaging," *Monthly Weather Review - MON WEATHER REV*, vol. 138, 01 2010.

[108] E. M. L. Beale and R. J. A. Little, “Missing values in multivariate analysis,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 37, no. 1, pp. 129–145, 1975.

[109] S. Buck, “A method of estimation of missing values in multivariate data suitable for use with an electronic computer,” *Journal of the Royal Statistical Society. Series B*, vol. 22, 07 1960.