MACHINE LEARNING MODELS APPLIED TO ENERGY TIME-SERIES FORECASTING

PH.D. ON ELECTRONICAL, COMPUTER SCIENCE AND ELECTRICAL ENGINEERING

XXXII CYCLE

SUPERVISOR: Prof. Giuseppe De Nicolao
AUTHOR: Andrea Marziali

UNIVERSITÀ
DI PAVIA

Abstract

La rivoluzione digitale iniziata negli anni '50 con l'introduzione del primo computer digitale ha ormai pervaso quasi ogni attività comportando in larga parte il loro cambiamento. Una delle principali conseguenze della digitalizzazione è certamente data dalla mole di dati a disposizione che oggigiorno è molto superiore rispetto al passato, per questo fanno ormai parte del gergo comune parole come Big Data. Uno degli ambiti fortemente influenzati dall'aumento della disponibilità di dati è quello delle previsioni, dove le previsioni *data-driven*, basate tipicamente su modelli di *machine learning*, hanno affiancato e, in alcuni casi, sostituito i modelli statistici precedentemente primariamente utilizzati. Almeno per quanto riguarda il caso specifico trattato in questa tesi, quello della previsione di serie temporali nel settore dell'energia, l'evoluzione modellistica ha fatto un passo ulteriore: per poter aumentare l'accuratezza delle previsioni, spesso la soluzione non è quella di basarsi su un solo modello ma sviluppare modelli aggregati che cioè combinano le previsioni di vari modelli base. Questa duplice evoluzione modellistica, seppur volta ad un miglioramento delle performance in termini di accuratezza delle previsioni, porta con sé una maggiore complessità nei modelli utilizzati ed una conseguente diminuzione nella comprensione ed interpretabilità del risultato.
Nel corso dei tre anni di dottorato ho svolto un lavoro di ricerca incentrato sull'applicazione delle tecniche di machine learning (ML) alla previsione di serie temporali nel settore dell'energia, in particolare la domanda gas e la domanda elettrica in Italia. Le previsioni sono relative al giorno successivo alla data di previsione (one-day-ahead). I dati a disposizione per le analisi sono stati la serie della domanda gas italiana, disaggregata nei tre settori principali, civile (RGD), industriale (IGD) e termoelettrico (TGD) dal 2007 al 2018, e la serie della domanda elettrica italiana (IED) dal 2012 al 2018.
Le previsioni day-ahead sono state ottenute attraverso l'implementazione in Python di nove modelli di ML – tre modelli lineari regolarizzati (regressione ridge, LASSO, elastic net), un processo gaussiano (GP), una support vector machine (SVM), un k-nearest neighbors (KNN), una rete neurale feedforward (ANN), un modello dato dalla combinazione lineare di funzioni sinusoidali su dominio toroidale (torus) ed infine una foresta di alberi (random forest) – e di cinque diversi metodi di aggregazione (modelli ensemble) – (i) media semplice, (ii) media pesata, (iii)-(iv) due modelli (Subset average (c.a.) e Subset average (b.f.)), in cui si fa la media semplice su un sottoinsieme delle previsioni dei modelli base, ottenuto attraverso due differenti processi di ottimizzazione, (v) una SVM con input dati dalle previsioni dei nove modelli base. È stato altresì proposto un metodo per l'identificazione di tre reti neurali ricorrenti (RNN) – una RNN semplice e due architetture più complesse, Long-Short Term Memory (LSTM) e Gated Recurrent Unit (GRU) – utilizzate per la previsione delle medesime serie temporali.
Particolare attenzione è stata posta sul valore aggiunto dato dall'aggregazione delle previsioni sia nel caso di modelli ensemble, che nel caso dell'aggregazione data dalla semplice somma o media di previsioni più di dettaglio. Esempi di questo sono: (i) la previsione della domanda elettrica italiana, ottenuta attraverso la previsione delle singole ore per le diverse zone di mercato, (ii) le previsioni finali delle RNN, ottenute come media delle diverse previsioni ottenute variando il punto iniziale nello spazio dei parametri.

Se l'apparato modellistico sviluppato può essere applicato anche a serie relative ad altri paesi del mondo, la scelta e le modalità di utilizzo di alcuni regressori, come per esempio la previsione della temperatura, sono cruciali per la correttezza delle previsioni ma più tipiche dell'Italia, come conseguenza della sua posizione geografica e delle sue tipicità nell'utilizzo di gas ed elettricità. In particolare, la temperatura nella stagione fredda, sotto i 18°C, influenza fortemente la domanda gas civile portando la stessa a valori anche quattro-cinque volte più alti rispetto a quelli estivi dove l'effetto della temperatura è assente. Questa forte dipendenza dalla temperatura inserisce nella domanda gas civile una rilevante componente stocastica legata alla previsione one-day-ahead della temperatura stessa. Data la sua importanza per la previsione della domanda gas civile, attraverso un semplice modello, validato anche empiricamente, è stato possibile quantificare l'impatto dell'errore della previsione della temperatura sull'errore complessivo nella previsione della domanda gas. Grazie a tale modello è stato anche possibile calcolare un limite inferiore per l'errore che si può commettere nella previsione della domanda gas, ponendo a zero tutte le altre fonti di errore. Tale limite inferiore è servito per valutare la bontà dei modelli testati per la previsione di RGD.
Dai risultati ottenuti si è verificato che non esiste un modello nettamente migliore degli altri per tutte le serie da prevedere ma che i modelli migliori, in genere, quasi si equivalgono in termini di performance, con una leggera supremazia da parte delle reti neurali ricorrenti. D'altra parte i risultati mostrano chiaramente che, per migliorare la loro accuratezza, è meglio ricorrere a tecniche di aggregazione e non limitarsi ad un singolo modello. Altro punto chiave, avvalorato dai risultati ottenuti, è dato dalla scelta dei fattori, e quindi dei dati in input ai modelli, che hanno rappresentato la fonte principale di miglioramento delle previsioni: esempio principe in questo ambito è dato dall'inserimento della previsione one-day-ahead della temperatura, fattore cruciale per la previsione sia della domanda gas che di quella elettrica, che, a seconda della serie oggetto di previsione e del modello utilizzato, va trasformato opportunamente al fine di massimizzare le performance.

"Prediction is very difficult, especially if it's about the future."

Niels Bohr

I dedicate this work to my loved ones.

Contents

1

Introduction

Time-series forecasting is the main topic of my thesis. It consists of predicting future values of series whose elements are ordered in time and, consequently, serially correlated. Time-series forecasting is a wide field of research because, in almost any context, people, based on the past and present, are interested in predicting the future in order to take some appropriate action. Moreover, the different extents of the forecast horizon, short, medium, and long, contribute to enlarge this field of study. A major classification of the methods adopted is between the qualitative and the quantitative ones. Qualitative methods are mostly used in case of medium- and long- range predictions, also because other considerations, complementary to the past and present data, are evaluated in order to cope with the vast possible scenarios. Instead, quantitative methods [1] are the cornerstone of short-term forecasting, where past and present data represent the principal source of knowledge.
In particular, this work is focused on the short-term, one-day-ahead, forecasting of energy time series of gas demand and electricity loads.
Nothwistanding the limited perimeter of the work, the relevant literature is huge as detailed in the review articles [2], [3] for gas demand and [4], [5], [6], [7], [8] for electricity loads. For both sectors, in the past the principal forecasting methods used to be linear regressions and time-series or Box-Jenkins models. The rise of machine learning and statistical learning theory as a framework for data-based prediction, revolutionized prediction algorithms, opening the way to the widespread adoption of neural network approaches [9], [8].
In this thesis, the first problem addressed is that of one-day-ahead gas demand forecasting through the prediction of its three main components (residential, industrial, and thermoelectric). The analysis of the different time series of gas and electricity reveals that they present similar characteristics, so that most of the regressors, employed for the three gas demand series, were afterwards used also for electricity load forecasting. Of particular interest is the relationship, also non-linear, of these time series with the temperature. In order to capture this relationship directly from the data, different statistical learning models were applied and their performances assessed by varying the time series used for calibration and test. All the forecasters, both for gas and electricity, were developed with reference to the Italian data: in case of gas demand, the available data covered the years between 2007 and 2018, whereas for electricity the data ranged from 2012 to 2018.
The main contributions of the thesis can be grouped in three main areas.

The first one regards the study of the influence of weather forecast errors on natural gas demand models. Despite being critical in industrial applications, previous works seldom specify if the predictors use forecasted or observed temperature, maybe due to the belief that temperature errors have negligible impact. In contrast, for residential gas demand, which shows the most evident dependence on temperature, in chapter 4 the influence of weather forecast errors is investigated, both theoretically and experimentally. A novel easy-to-compute bound is derived that predicts the best achievable root mean square error (RMSE) as a function of the temperature RMSE. This bound is then validated on experimental data: Italian RGD forecasts are obtained using both observed and predicted temperatures, thus allowing for a quantitative assessment of accuracy degradation.
The second main contribution is the benchmark of a number of statistical and machine learning forecasters that were subjected to a rather unique validation, both for duration and variety, performed on several years of Italian electricity loads and gas demand (disaggregated into its three main components). Nine prediction models were considered: ridge regression, LASSO, elastic net, Gaussian Process, support vector regression, nearest neighbours, Artificial Neural Networks, torus model, random forest. Moreover, built on the nine base models, five ensemble predictors were considered: simple average, weighted average, support vector regression aggregation and two subset average methods, where the subset of base predictions is chosen by an appropriate optimization process. In the case of electricity demand, also the aggregation of finer (either spatially or temporally) forecasts was investigated, by comparing the direct forecast of daily Italian electricity loads with the forecast of the same quantity obtained by summing the hourly or the zonal-hourly forecasts.
The third main issue was the implementation and testing of three Recurrent Neural Networks (RNN) models - Long Short Term Model (LSTM), Gated Recurrent Unit (GRU) and simple Recurrent Neural Network (RNN0) - for the day ahead forecasting of the Italian gas and electricity demands. Special attention was given to the model identification phase, including the choice of hyperparameters and the impact of random initializations.

At the end of the thesis, a summary of the notation used is reported together with the list of abbreviations.

1.1 Publications and reports

- Alice Guerini, Andrea Marziali and Giuseppe De Nicolao (2018). "MCMC calibration of spot-prices models in Electricity markets." Published in Applied Stochastic Models in Business and Industry.[10]

- Emanuele Fabbiani, Andrea Marziali and Giuseppe De Nicolao (2018). "Fast calibration of two-factor models for energy option pricing." arXiv preprint arXiv:1809.03941.

- Andrea Marziali, Emanuele Fabbiani and Giuseppe De Nicolao (2019). "Forecasting residential gas demand: machine learning approaches and seasonal role of temperature forecasts." arXiv preprint arXiv:1901.02719. Accepted for publication in International Journal of Oil, Gas and Coal Technology.

- Andrea Marziali, Emanuele Fabbiani and Giuseppe De Nicolao (2019). "Ensembling methods for countrywide short term forecasting of gas demand." arXiv preprint arXiv:1902.00097. Accepted for publication in International Journal of Oil, Gas and Coal Technology.[11]

- Andrea Marziali and Elisa Raspanti (2019). "Italian short term load forecasting: different aggregation strategies." Submitted to Journal of Forecasting, now under review.

All these papers are related to my PhD studies, but only the last three cover topics included in the present PhD thesis. The contents of these three papers are largely reported in chapters 4 and 5.

2

Electricity and gas: demand and prices

2.1 Introduction to the energy sector

The XX century was full of events in the energy sector. It started with the diffusion of cars and the correlated increase of oil consumption to produce gasoline and fuels used by the same vehicle engines. This century laid also the basis for the new renewable energies by the constructions of the first geothermal plant, wind turbines, photovoltaic cells, tidal power plants and also first experiments regarding the production of wave energy. Finally, nuclear energy was discovered and the first nuclear power plants were built.
The last part of the XX century, as well as the beginning of the XXI, brought a growing attention to the issues of environmental impact and global warming. Therefore, nuclear energy and, above all, renewable energies became preferred to fossil ones due to lower pollution impact and carbon emissions. Nevertheless, fossil fuels are still the most used at world level: oil, followed by coal and gas cover about the 80% of world energy consumption, the other 20% being given by nuclear and renewable energies. China is the biggest producer and consumer of energy overall, and above all of coal, followed by the USA, which is the first in terms of oil and natural gas production. Regarding the electricity production, the energy sources are fossil for about 65% (coal 39%, gas 23% and oil 3%), nuclear for a little more than 10%, and the renewable sources represent the remaining 25%. In the last thirty years, the coal percentage has been stable whereas the oil one, reduced of about 8%, has been substituted by the gas one which has increased of about 9%, with the consequent decrease of total carbon emissions. Nuclear consumption has been decreasing from the peak of 1996, when it covered the 17% of the total production, whereas the hydroelectric component is stable. The XXI century shows a steep rise in other renewable sources, mainly wind, biomass, and photovoltaic, which increased from 1% to 9%.
In Italy, the breakdown by energy is similar to the world one, with 80% from fossil sources (gas 40%, oil 34%, and coal 6%) and 20% from nuclear and renewable. The electricity production is thermoelectric for 71%, mainly from gas and coal, increasing from the minimum registered in the year 2014 after the decline started in 2007 when it covered 85%. The hydroelectric component is stable at 13%; the photovoltaic production began in the 2007 and now is around the 8%, whereas the wind power started its development at the beginning of this century and now reached about 6%. Finally, the geothermal generation is stable at about 2%.

2.2 General description of Italian Energy markets

As a consequence of the European directive about the internal energy market development (96/92/CE), the Italian electricity market was born with the legislative decree number 79 (79/99) of 1999, on the 16th of March. The aim at developing an electricity market is twofold: promoting the competition in power production and sale, by the creation of a marketplace, enhancing the transparency and objectivity, and fostering the highest efficiency and transparency in the electricity dispatching, managed as a natural monopoly. On 1st April 2004, the first negotiations on the Italian Power Exchange were made. At the beginning the only spot market was introduced, while on 1st November 2009 the electricity forward market was also started.
The principal roles in the Italian electricity market are played by four different entities. First of all, there is *Gestore dei Mercati Energetici* (GME) which manages the energy markets, both for electricity and gas, maximizing the targets of their constitution such as transparency, neutrality, objectivity, and competition between producers. Terna S.p.A. manages the National Transmission Grid and the electricity flows by the dispatching activities, ensuring the balance between demand and offer at the hourly level. The role of guarantor of competition and efficiency by adjustment and control features is played by the Authority of electricity and gas (AEEG), whereas the Ministry of Economic Development is the owner of the strategic actions aimed at ensuring the sustainability of the Italian energy system.
The European directive 98/30/CE and the legislative decree number 164 of 2000, on the 23rd of May, so called "Decreto Letta", set the basis for the liberalization of the gas market. Such decree required the functional unbundling for what concerns the activities of transport and dispatching from all the other activities around the natural gas, later extended to the storage activities with the "Seconda direttiva gas" (2003/55/CE). Another focal point of the decree 164 was the setting of the limits on the maximum quantities introduced for sale in the Italian system and limits on the sales to final customers. Many other legislative decrees in the years went in the direction of greater competition in the natural gas market and ruled its change until nowadays. In this context, the platform for the natural gas exchange market in Italy was born with the decree of May, 2010.
Analogously to the Italian electricity market, AEEG plays the role of guarantor of competition and efficiency by adjustment and control features in the Italian natural gas market as well. Snam Rete Gas (SRG), as the Transmission System Operator (TSO), is the principal owner of gas transmission, dispatching and guarantor of the system balancing. Moreover SRG supplies the different market participants with intraday reports about statistical data on gas market. The exchange platform, just as the electricity one, is managed and organized by GME that has also the role of central counterparty.

2.2.1 Electricity market

The Italian electricity market is divided into two parts: the spot market and the forward market. The spot market is divided into several sessions grouped in Day-Ahead Market (MGP), Intra-Day Market (MI), and finally, Ancillary Services Market (MSD). Each of these market sections is necessary to match at the best final offer and demand in real-time. The MGP and MI sessions are based on zonal-hourly auctions, and each of them ends with a single system marginal price (SMP). On the other hand, in the MSD sessions, the price is set as a pay-as-bid mechanism.

The spot market sessions start with the MGP in the day before the delivery date and continue with seven sessions of price adjustment (MI) temporally divided in the day before and on the same delivery date. In the same period, there are different MSD sessions where Terna, such as a central counterparty, requires to the operators to reduce or enhance their production, with respect to what planned in the MGP and MI sessions, accepting their bids.
In the forward market, opened every day, GME plays the role of central counterparty and operators can buy or sell contracts with future deliveries on month, quarter or year periods. Alternatively, operators can exchange the same contracts over the counter (OTC). Each forward contract is registered in the *Piattaforma dei conti energia* (PCE), a mechanism introduced to give more flexibility to the operators separating the phase of contract registration from the following physical programs registration. By the PCE, the exchanged energy in the forward contracts takes part to the MGP auction to concur to the zonal-hourly system marginal price.
A more detailed analysis of MGP, the first and principal spot market, can highlight the importance of demand forecasting for the electricity players in order to maximize their revenues. All the description of the MGP mechanism to arrive at the equilibrium price applies, exactly in the same way, to the different zones at hourly level.
The MGP market starts with offers by the participants, where they indicate a maximum/minimum price they can buy/sell a fixed volume of electricity. In order to not unbalance the electricity system, operators and consumers in the real-time have to respect their final program after all the sessions of the market, otherwise they incur in penalties. This represents a guarantee also in relation to the correctness of the offers in MGP respect to the real-time operations. In this context, in order to maximize their revenues, a crucial information for the electricity sellers is the hourly consumption of the day ahead, which is not known but could be only predicted. The MGP mechanism, to accept the offers, orders all the offers to sell with ascending price (offer curve) and the offers to buy with descending price (demand curve), as shown in fig. 2.1. Being the two curves aggregated in terms of quantities, the intersection between them yields the equilibrium price and the total exchanged quantity. Important constraints to be satisfied are the limits on flows of electricity between the neighbouring zones; in case these ones are respected, the equilibrium price is one for all the zones, while when, for a zone, the transmission capacity limits are saturated, the equilibrium price is fixed for that zone and the auction mechanism is repeated for the other zones to end with two or more prices among the different Italian zones. At the end of MGP auctions, GME determines the single national price (PUN) for the exchange of electricity in Italy for the day ahead.

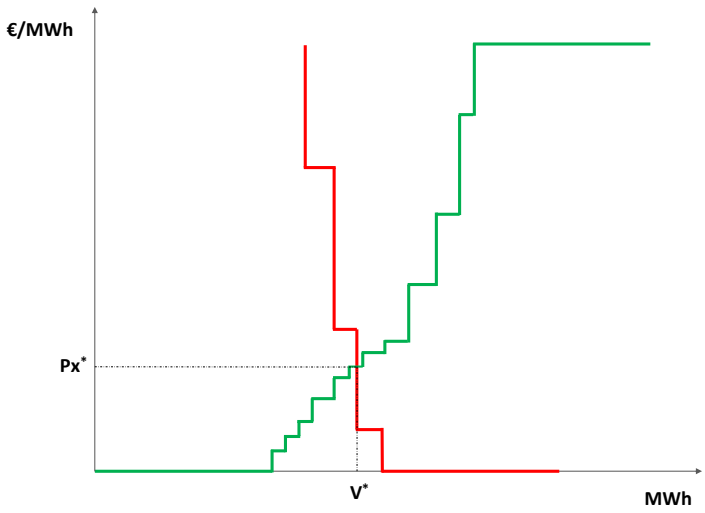


Fig. 2.1: A stylized representation of the mechanism of price determination in the MGP market. The intersection between the demand curve (red line) and the offer curve (green line) brings to the equilibrium price Px* and the total accepted demand V*.

It is clear, from the description of the auction mechanism behind the formation of the MGP zonal-hourly price, the fundamental role played by the consumption so that, for the energy companies, better predictions are extremely helpful to improving their bidding strategies in a more accurate way.

2.2.2 Gas market

Since October 2003, the operators can exchange gas in the Italian network at a virtual point, the "Punto di Scambio Virtuale" (PSV). In this point, very useful for the system balancing, the transactions are made over the counter by bilateral contract.
In addition to the OTC bilateral contracts there are ruled gas markets, owned by GME, among the operators authorized to make transactions at the PSV. Similarly to the electricity ones, these markets, called MGAS, are divided into spot (MP-GAS) and forward (MT-GAS). The two principal sessions of spot markets are the day ahead (MGP-GAS) and intraday (MI-GAS) ones. Both are based on a pay-as-bid mechanism rather than an auction one, typical of electricity. The transactions on MGP-GAS, the most significant GME spot gas market, start three days before the delivery date and close at hour 2:30 of the delivery date. Then, the single price of the delivery date is given by the average of all the transaction prices weighted with the exchanged transaction quantities. This daily price is relevant because, as a consequence of recent regulations (since October 2016), it is also at the base of the balancing price which the operators have to pay for their unbalanced positions, that is the difference between the consumption nominated the day before and the true consumption of the delivery day. In order to obtain the balancing price, the other two relevant quantities regard the purchase and sales bids of the TSO

SRG, that participates to the gas market in order to assure the balance of itself. Indeed, as a consequence of the dual mechanism, any operator, in order to balance his position, receives

$$P^{sell}_{unbal,G} = \min\left(\min\left(P^{TSO}_{G}\right); \bar{P}_G - SA_G\right) \tag{2.1}$$

in case its actual consumption is higher than nominated (long position); whereas it pays

$$P^{buy}_{unbal,G} = \max\left(\max\left(P^{TSO}_{G}\right); \bar{P}_G + SA_G\right) \tag{2.2}$$

in case its actual consumption is lower than nominated (short position). In both the equations SA_G is a fixed small adjustment equal to 0.108 €/MWh, $\bar{P}_G$ is exactly the price MGAS of the considered day and P^{TSO}_{G} are the prices exchanged by SNAM. In particular situations, caused by missing or excess of gas, the balancing price respectively to pay or receive are 82.8 €/MWh and 0 €/MWh; whereas in case of scarce liquidity, exchange titles less than 2000 MWh, the prices are given by the average of the previous 30 day.
It goes without saying that, especially in order to correctly nominate the day-ahead gas consumption so as to limit the costs given by unbalanced positions, but also aimed at better forecasting the gas price, the availability of accurate predictions of the day-ahead gas demand is of great value for energy companies.

2.3 Italian Gas consumption

In this thesis, the prediction of the day-ahead Italian gas demand (GD) is addressed passing through its disaggregation. Apart from minor components that can be neglected, at any day t, the overall GD is given by the sum of Residential Gas Demand (RGD), Industrial Gas Demand (IGD) and Thermoelectric Gas Demand (TGD):

$$\mathrm{GD}_t = \mathrm{RGD}_t + \mathrm{IGD}_t + \mathrm{TGD}_t$$

RGD represents the main part of the overall Italian gas consumption, accounting for household usage for cooking, water heating and, most importantly, environment heating; IGD includes demand by industrial plants, while TGD only accounts for the fuel required by thermoelectric power plants.
Four one-day-ahead forecasting problems will be considered: (i) RGD, (ii) IGD, (iii) TGD and (iv) the overall GD forecast which will be obtained by summing (i), (ii) and (iii). The profiles of RGD, IGD, TGD and GD from 2007 to 2018 are respectively displayed in figs. 2.2, 2.3, 2.4 and 2.11.

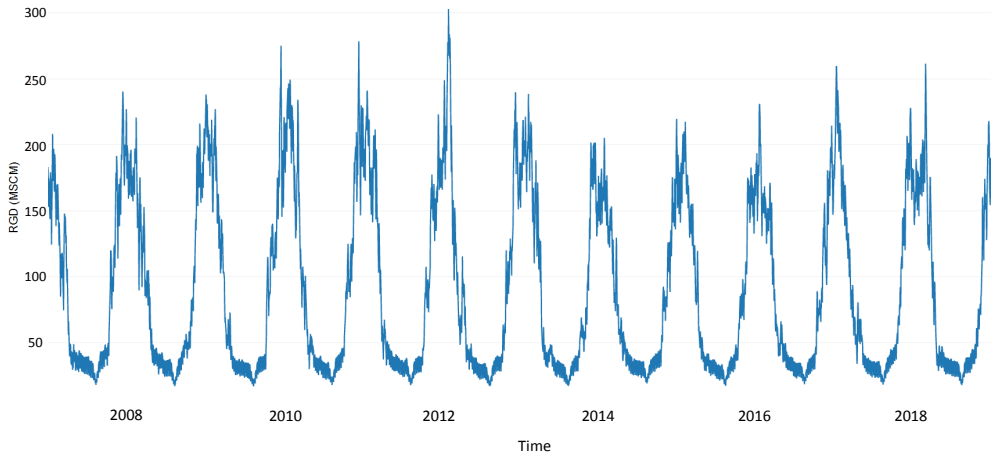


Fig. 2.2: Italian Residential Gas Demand (RGD): years 2007-2018

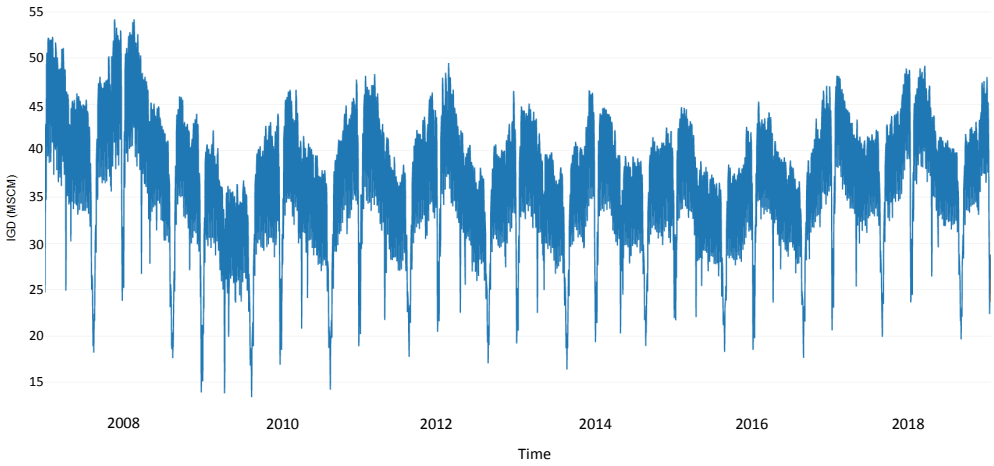


Fig. 2.3: Italian Industrial Gas Demand (IGD): years 2007-2018

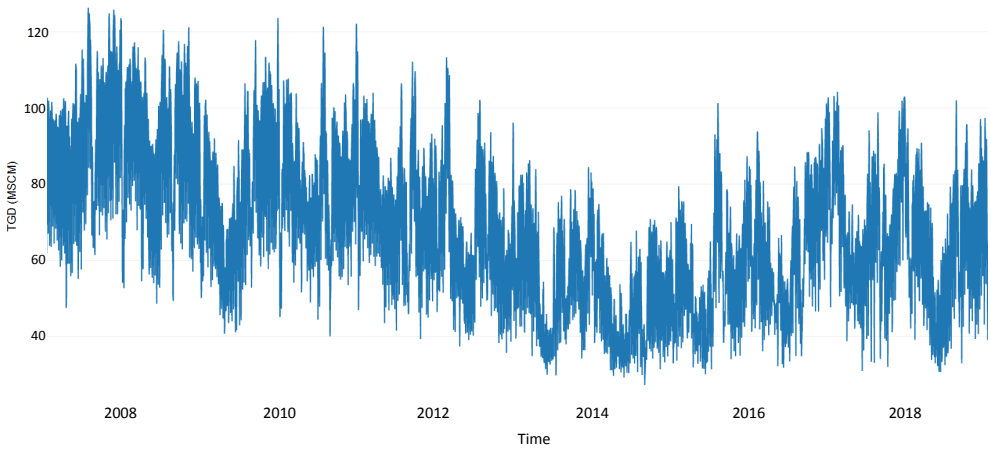


Fig. 2.4: Italian Thermoelectric Gas Demand (TGD): years 2007-2018

RGD, see fig. 2.2, greatly oscillates with the season: during the cold months, from October to March, it represents about 55% of the overall Italian demand, while it drops to about 27% during the warm months, from April to September. When the temperature climbs above 17-18 Celsius degrees, domestic heating is typically switched off. Thus, during the cold period lower temperatures cause a larger RGD, while, during summer, when weather influence is negligible, a seasonal pattern becomes evident, with lower RGD during weekends compared with working days. Due to the lack of dependence on weather conditions, the profile of summer RGD is remarkably repeatable from year to year. Overall, there is an evident yearly seasonality.
Industrial Gas Demand (IGD), fig. 2.3, does not exhibit strong trends: a significant decrease is only recorded in 2009, following the financial crisis started in the previous year. The series presents weekly and yearly seasonal patterns. In particular, as most of the industrial facilities stop or slow down production during the weekend, IGD is lower on Saturdays and Sundays. In August and at the end of December, regular holiday periods, IGD drops to about half of its average value. Other holidays, such as Easter and Labour Day, result in similar effects. During the year, IGD shows a decrease from January to August and an increase from September to December, due to the use of gas for environmental heating.
Differently, from RGD and IGD, TGD shows a clear trend, see fig. 2.4. From 2008 to 2014 TGD decreases, mostly due to the growing importance of renewable sources of electric power, while, since 2014, the trend stabilizes, likely due to the decrease in subsidies to the installation of photovoltaic systems. The yearly periodicity for TGD is less evident than for RGD and IGD.
In order to characterize the yearly seasonality, it is convenient to introduce the notion of similar day, widely used in this thesis. The following definitions hold:

- year(t) is the year to which day t belongs;

- weekday(t) is the weekday of day t, e.g. Monday, Tuesday, etc;
- yearday(t) is the day number within year(t) starting from January 1, whose *yearday* is equal to 1.

Definition 2.1 (Similar Day). *If t is not a holiday, its similar day $\tau^* = \mathrm{sim}(t)$ is*

$$\tau^* = \arg\min_{\tau} |\mathrm{yearday}(\tau) - \mathrm{yearday}(t)|$$

subject to

- $\mathrm{year}(\tau) = \mathrm{year}(t) - 1$*;*
- $\mathrm{weekday}(\tau) = \mathrm{weekday}(t)$*;*
- τ *is not a holiday.*

If t is a holiday, its similar day $\tau^* = \mathrm{sim}(t)$ is the same holiday in the previous year.

According to the Italian calendar, holidays are 1 January, 6 January, 25 April, 1 May, 2 June, 15 August, 1 November, 8, 25 and 26 December, Easter and Easter Monday. Following the Similar Day definition, the superposition of the twelve years of RGD, IGD and TGD are displayed respectively in figs. 2.5, 2.6 and 2.7. From these figures, the main features of the three time series can be better appreciated, above all the strong periodicities: yearly for both RGD and IGD, weekly for IGD all the year long and also for RGD but only during the central period of the year, when temperatures rise over 18°C.

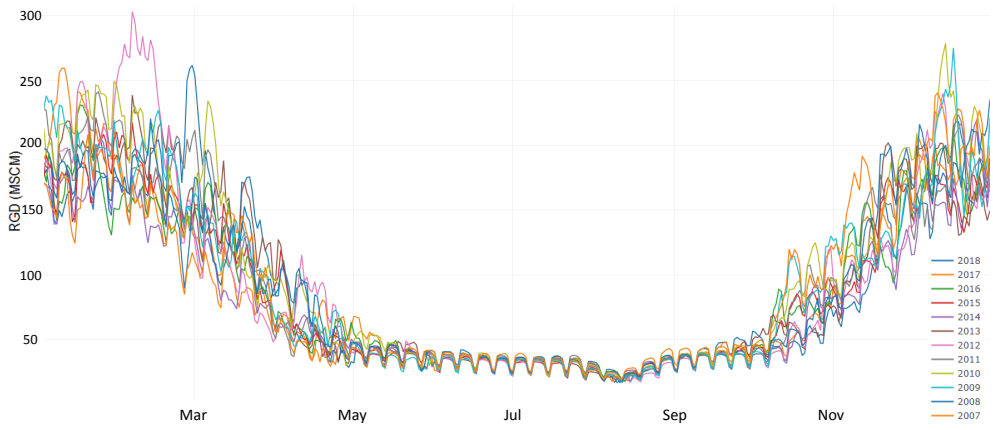


Fig. 2.5: Superposition of Italian Residential Gas Demand (RGD): years 2007-2018

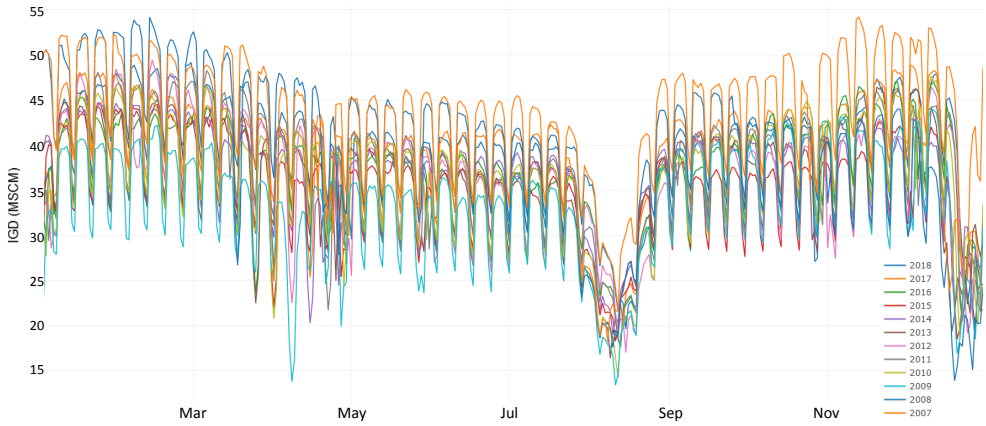


Fig. 2.6: Superposition of Italian Industrial Gas Demand (IGD): years 2007-2018

TGD shows a greater variability compared to IGD and RGD, as seen from the year-over-year plot in fig. 2.7. TGD is indeed influenced by several factors, including prices of electric power, gas, and European Emission Allowance (EUA) certificates, which exhibit a large volatility [12]. These observations explain why yearly periodicity is relatively less important in TGD than in RGD and IGD.

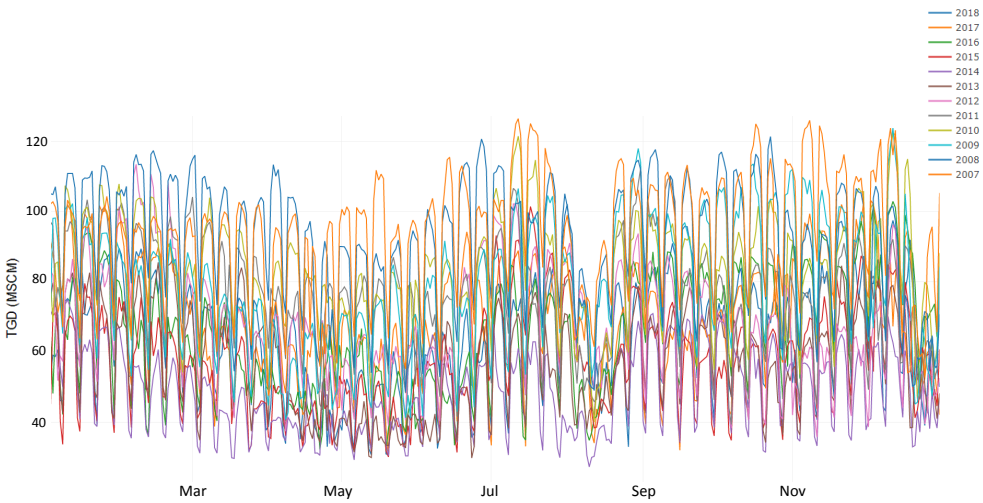


Fig. 2.7: Superposition of Italian Thermoelectric Gas Demand (TGD): years 2007-2018

Averaging the values of gas demand for similar days in the twelve years for each of the three segments (figs. 2.8, 2.9 and 2.10) removes the variability from the three time series. This operation has the effect of bringing up the weekly periodicity that in the case of TGD and of RGD was hidden by the intense volatility during the cold months.

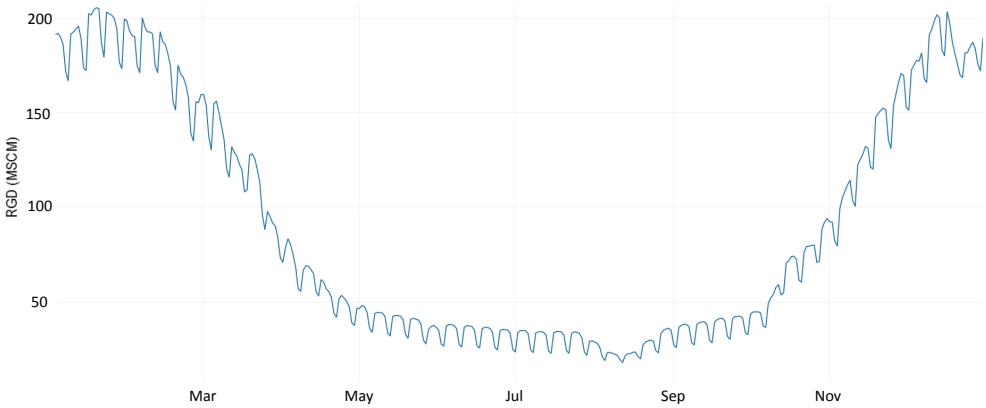


Fig. 2.8: Average profile of Italian Residential Gas Demand (RGD): years 2007-2018

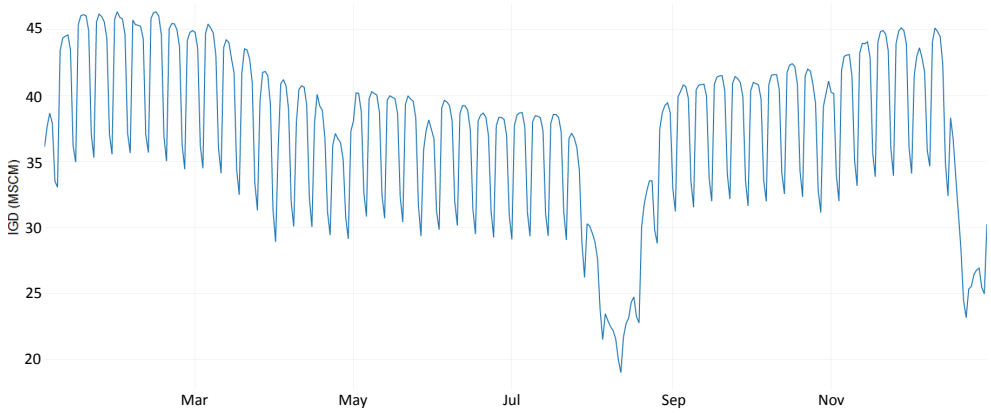


Fig. 2.9: Average profile of Italian Industrial Gas Demand (IGD): years 2007-2018

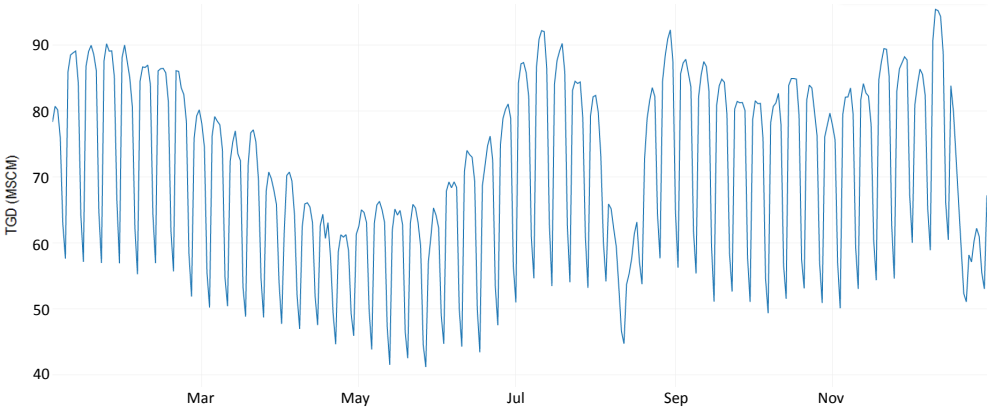


Fig. 2.10: Average profile of Italian Thermoelectric Gas Demand (TGD): years 2007-2018

In fig. 2.11 the GD time series is displayed. Over the twelve years 2007-2018, the domestic demand accounted for about 45% of the total consumption, while 35% was due to the thermoelectric component. It has to be noted that the composition of the daily demand

greatly differs between the winter season (from October 1 to March 31) and the summer one (from April 1 to September 30). RGD represents about 55% of the total demand in winter, but only 27% in summer. In summer the thermoelectric covers the biggest portion with the 48% as a consequence of his dependence on power price, whereas in winter it adds up to the 29%. The industrial component is always the lowest, 25% in summer and 16% in winter. This variability in the proportions of the three main segments depends on the pronounced difference between summer and winter in the residential gas demand. As already observed, during winter lower temperatures cause a larger RGD, so that the RGD yearly profile follows the seasonal temperature profile, see fig. 2.5.
All these characteristics sum up in the GD curve (figs. 2.11, 2.12, 2.13) which shows the typical yearly shape of RGD, but larger differences between the levels of the working days and those of weekends, just as for IGD and TGD. Moreover, the GD series exhibits higher volatility than RGD, derived from the TGD component, and a pronounced reduction during the summer and Christmas holidays, mostly following the IGD and TGD profiles.

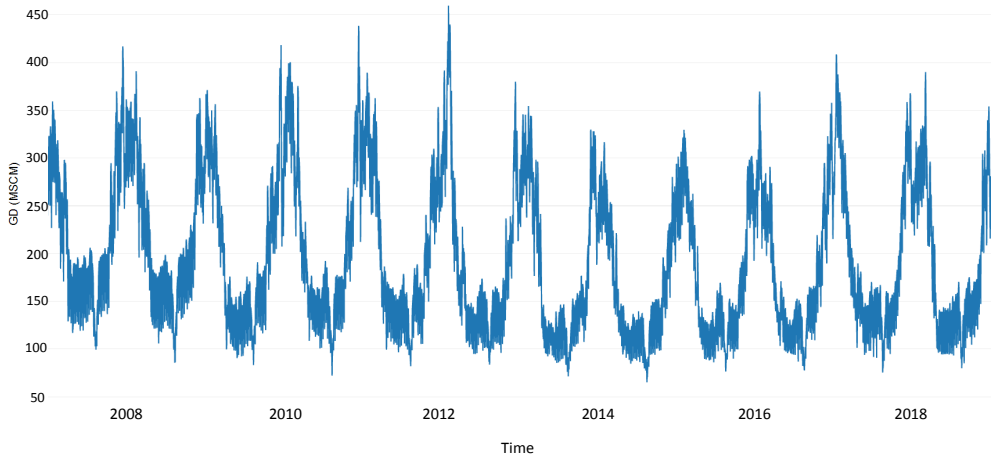


Fig. 2.11: Italian Gas Demand (GD): years 2007-2018

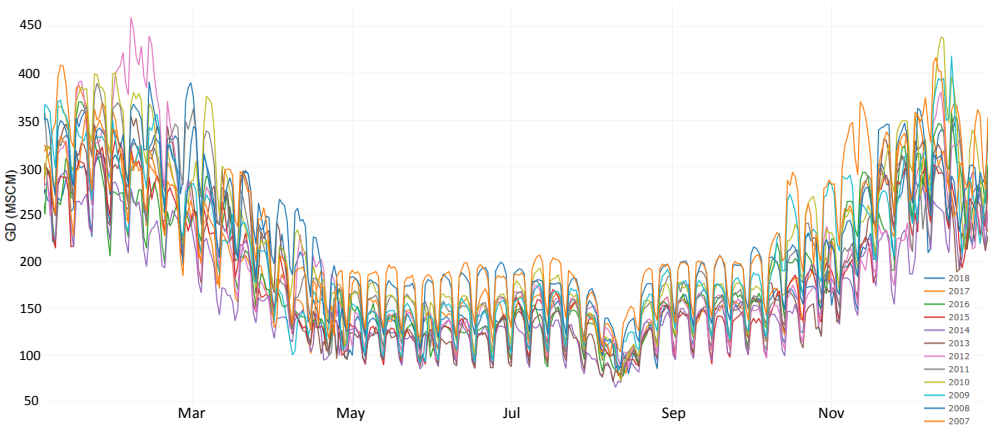


Fig. 2.12: Superposition of Italian Gas Demand (GD): years 2007-2018

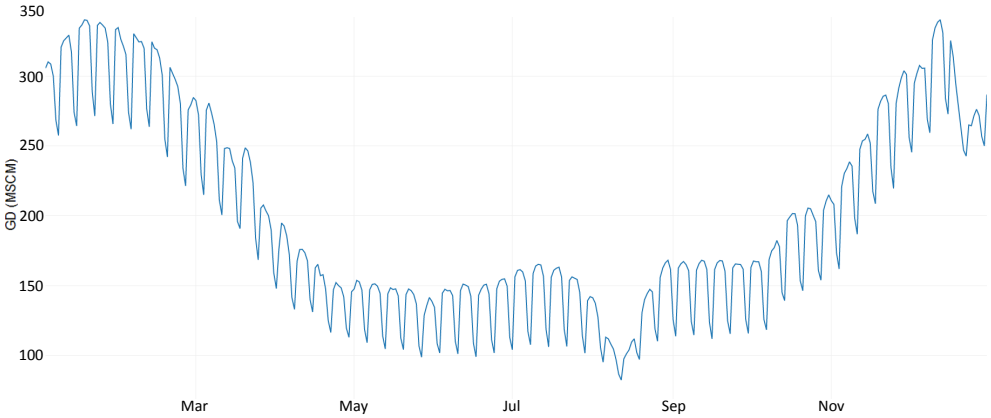


Fig. 2.13: Average profile of Italian Gas Demand (GD): years 2007-2018

2.4 Italian electricity load

In this thesis, also the day-ahead prediction of the Italian electricity demand (IED) is addressed both at aggregated and disaggregated level.
The IED time series during the period 2012-2018 is displayed in fig. 2.14. Its characteristics are similar to those highlighted in the Italian gas demand series, in particular the pronounced seasonalities and some correlation with the temperature.
IED exhibits daily and weekly seasonalities. As a matter of fact, in the period 2012-2018, the average percent differences between consecutive days are characterised by a steady growth of 28% between Sunday and Monday, low variations between the following days until Friday and two pronounced decreases, between Friday and Saturday and between Saturday and Sunday, respectively of 15% and 12%. The weekly periodicity is the most evident. Another periodical signal is shown by the biannual raising of the demand level in correspondence of winter and summer, generally in February and July. A third periodicity comes from the reduction of the demand level during the holiday periods, generally three times a year, around the end of April and the beginning of May, in August and during the Christmas period.

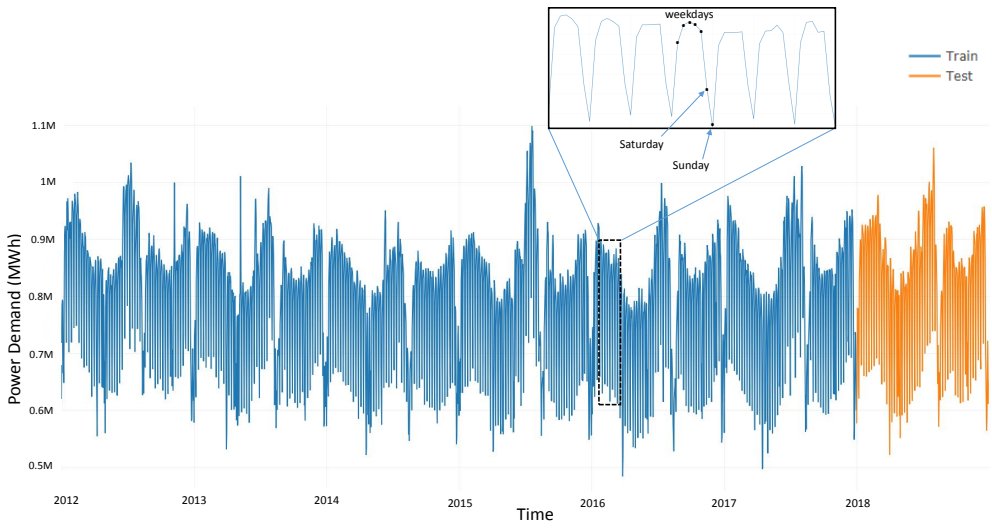


Fig. 2.14: The daily Italian electricity demand in the period 2012-2018.

Fig. 2.15 shows the breakdown of the total daily IED (IED_d) in its 24 hourly components (IED_h) during 2018, the most recent sample year. The average correlation between the daily series of the different hours is high, about 0.82, with a minimum of 0.45. The lowest correlations refer to the series of the first six hours of the day, with mean 0.67 and minimum 0.45, whereas the other hours exhibit a mean of 0.94 and a minimum correlation of 0.78. On the other hand, the correlation between the 24 series IED_h and IED_d has a mean value of 0.9 and a minimum of 0.62. Here as well the result is highly diversified in

the two clusters of hours, the first six with a mean of 0.72 and a minimum of 0.62, and the other hours with a mean of 0.97 and a minimum of 0.91. In line with the high correlation between the different IED_h and with IED_d, the behavior of IED_d is roughly replicated by each IED_h only changing the level.

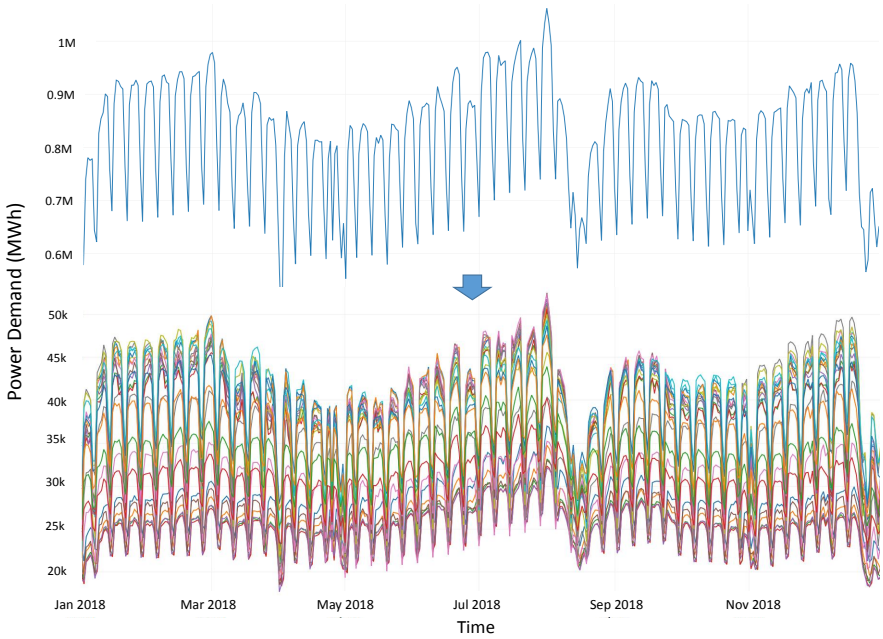


Fig. 2.15: IED_d versus time in the year 2018 (top) and its split into 24 daily curves IED_h (bottom). The figure highlights the similar behavior of each IED_h with respect to IED_d.

A further split of the IED series can be made at spatial level, leading to XED which accounts for the contribution of the six different zonal electric demands (fig. 2.16): North, Central-North, Central-South, South, Sicily, and Sardinia. XED turns into XED_d and XED_h respectively in case of daily and hourly demand.
As seen in fig. 2.17, the main contribution to IED is given by the North zone with about 56% of the total Italian demand, which is around 300 TWh. The main factors are the following: the North zone contains the highest number of regions: eight (all the regions north of Tuscany); they are much more industrialized compared to the other areas of Italy; moreover the climate has a significant effect because of the cold winter and hot summer. The Central-South zone, with three regions (Lazio, Campania, and Abruzzo), accounts for 15% and the Central-North (Tuscany, Umbria, and Marche) with 11% of the total IED. The South (Molise, Basilicata, Apulia, and Calabria) follows with 9% and the remaining part is due to the islands, with 6% and 3% respectively for Sicily and Sardinia. Another essential element to underline is the ratio between the mean of the weekly ranges (the difference between the highest and the lowest value of each week) and the average of the weekly mean levels. This ratio should follow the degree of industrialization of the region because it represents the relevance of the demand being higher on weekdays compared to

weekends when the industries are closed. As expected, the most significant value is found in the North, about 40%, followed by the Central-North with 32%, Central-South with 22%, South with 16%, Sicily with 12% and Sardinia with 8%.

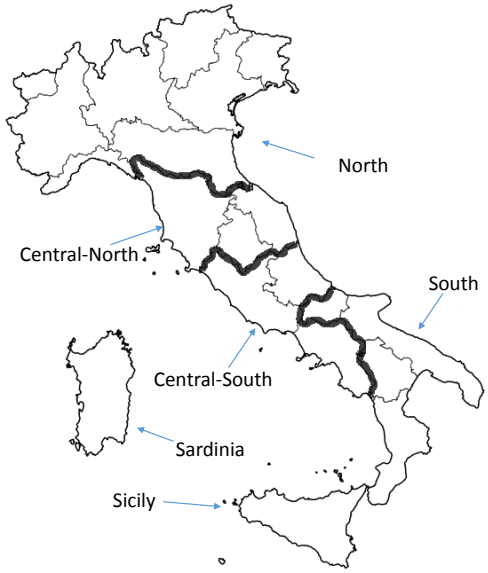


Fig. 2.16: The division of Italian territory into six market zones: North, Central-North, Central-South, South, Sicily, and Sardinia.

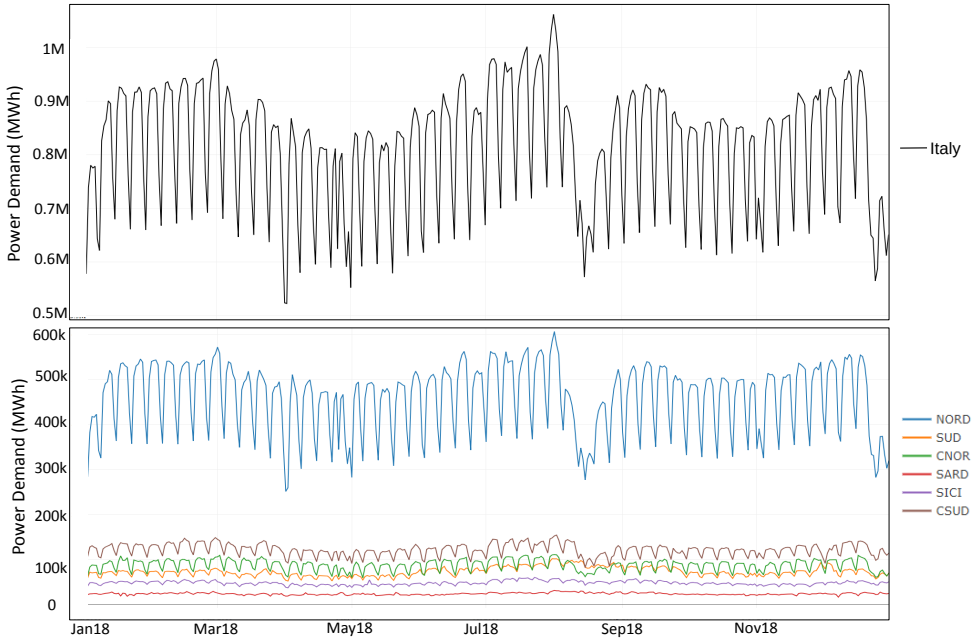


Fig. 2.17: Split of IED (top) in the six zonal contributions XED (bottom) for the sample year 2018. The North zone is by far the most important.

In fig. 2.18, six distinct plots with the 24 XED_h time series for each Italian zone are reported. Unlike the levels of the zonal demands, already discussed for fig. 2.17, here it is possible to observe for each zone the degree of regularity of the hourly paths. The North shows the most regular paths as also confirmed by the average correlation between consecutive samples of its XED_h, around 0.84. On the other hand, Sardinia XED_h follows the most irregular paths which involve a lower correlation of about 0.75. Nevertheless, all the XED_h exhibit a behavior similar to IED. This will be the main justification for the decision to forecast them with the same model and features chosen for IED forecasting.

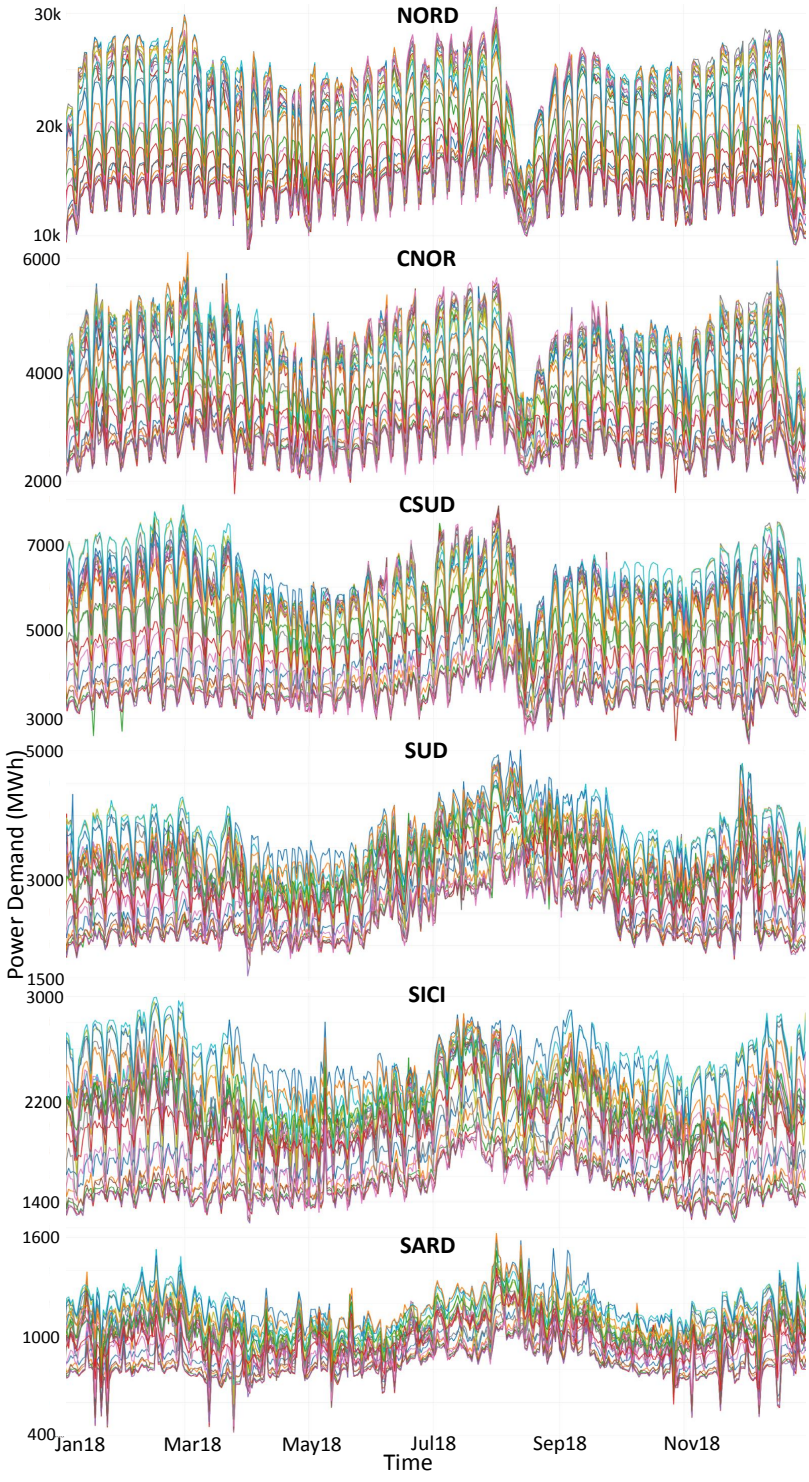


Fig. 2.18: The plots for the 24 XED_{h} time series for each Italian zone.

3

Machine Learning models

3.1 Forecasting models in general

The classical methods used for time series forecasting are linear Box-Jenkins models such as SARIMA, where the forecast is based only on past values of the time series, and SARIMAX, that accounts also for exogenous variables. A major drawback of classical linear models is given by discontinuities due to holidays and the possible presence of other nonlinear phenomena. In order to overcome these difficulties, herein the forecasting is formulated as a statistical learning problem.
The subject of this chapter is the overview of the *statistical learning* framework with a particular focus on models and methods of "learning from data" used in the thesis. The *statistical learning* framework is well represented by a tree (fig. 1) where the different branches end to the leaves where the predictive models are placed. Next, we review the break points of the tree, highlighting their motivation.

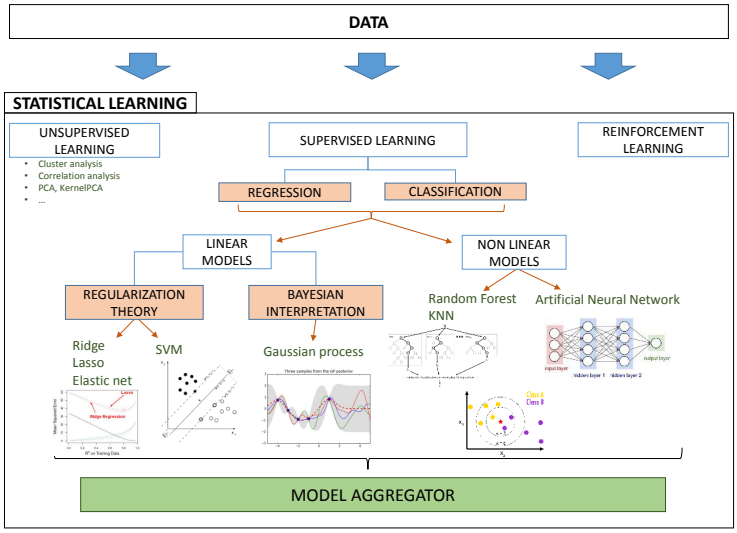


Fig. 3.1: A representation of statistical learning with a focus on supervised learning and the analyzed models in this thesis

Statistical learning starts with data, that represent by numbers what we want to learn, the target, and what we can use to learn, the features. The data are the practical examples from which we try to derive a rule able to predict the unobserved target based on the features. The typical situation is to have a set of pairs $\{\mathbf{x}_i, y_i\}$ where, in a forecasting context, i labels the time step and $\mathbf{x}_i$ is a vector gathering the features that we use to predict the target y_i. This branch of statistical learning, called *supervised learning*, is what is employed in this thesis to address short-term forecasting problems. On the other hand, there are situations in which the values assumed by the target variable are not known, bringing to another branch called *unsupervised learning*. The third area of statistical learning is given by the *reinforcement learning* which is based on the choice of the actions the machine has to take, in order to maximize a reward function.
Focusing on the *supervised learning* area, the first break point is created by the knowledge or not of the target variable; then, depending on the uncountable or countable nature of y, we have two further branches: *regression* or *classification*. All the models developed and tested in the thesis are *regression* models so, even if the following part of the tree comply with both *regression* and *classification*, we focus on the first one.
The simplest form of *regression* is based on *linear models*, where linear means a linear relation between the target and the regressors that could also be a non linear map $\phi(\mathbf{x})$ of the features $\mathbf{x}$ as long as their relation with the target is linear:

$$y = w \cdot \phi(\mathbf{x}).$$

Within the *linear* framework it is possible to describe many models, based on the different structures which the feature maps can assume. We will describe some parametric models, mentioning also regularization theory and Gaussian Processes, based on a Bayesian interpretation of the statistical learning framework.

On the other hand, we speak of nonlinear models when there is a nonlinear relationship between the target variable and the regressors, which, in the case of parametric models, happens when the target is a nonlinear function of the parameters $\mathbf{w}$. Among the many possible models, in this thesis we will describe three classes: *KNN*, *random forest* and *artificial neural networks* (ANN), the last one being the widest class.
A consequent and relevant part of the following sections will concern also the important role played by the ensemble methods given by the aggregation of forecasts in order to obtain better and more robust performances.

3.1.1 Importance of the predictive features, attention to overfitting

The objective of forecasting is to discover the value of the target variable y. In order to reach this scope, many regressors, possibly highly correlated with y, are searched. Thus the typical situation of supervised learning is to have n data samples $\{\mathbf{x}_i, y_i\}$, where $\mathbf{x}_i \in R^m$ is a vexctor of m features and $y_i \in R$ is the corresponding target variable y. These samples can be seen as n extractions from the unknown distribution $\Pr(\mathbf{x}, y)$. The goal of learning is to find a function $f(\mathbf{x})$ that minimizes the expected risk

$$\mathcal{E}(f) = \int_{\mathbf{X}\times Y} L(y, f(\mathbf{x}))d\rho(\mathbf{x}, y) \tag{3.1}$$

where L is the loss function that measures the difference, or error, between $f(\mathbf{x})$ and y, e.g. $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$, [13],[14].
Then, we can write the target y as

$$y = f(x) + \epsilon \tag{3.2}$$

where $\epsilon := y - f(x)$ is the so called irreducible noise. Based on a finite sample of data $\{\mathbf{x}_i, y_i\}$ (*training sample* or *in sample* - IS), it is not possible to find f but rather $\hat{f}$, which minimizes an empirical error. However, for a given loss function, the goal remains that of achieving a small expected loss on a new and different sample of data (*test sample* or *out of sample* - OOS). The problem is hence different from fitting, just because as OOS is different from IS. So the concept is that the predictor, based on a training sample, should enjoy good generalization properties, meaning that good performances are obtained also on the test sample. Assuming a L_2 loss function given by $L(y, \hat{f}(\mathbf{x})) = \left(y - \hat{f}(\mathbf{x})\right)^2$, its expectation value is given by

$$\begin{aligned}
\mathbb{E}[(y - \hat{f}(\mathbf{x}))^2)] &= \mathbb{E}[y^2 + \hat{f}(\mathbf{x})^2 - 2y\hat{f}(\mathbf{x})] \\
&= \mathbb{E}[y^2] + \mathbb{E}[\hat{f}(\mathbf{x})^2] - \mathbb{E}[2y\hat{f}(\mathbf{x})] \\
&= Var[y] + (\mathbb{E}[y])^2 + Var[\hat{f}(\mathbf{x})] + (\mathbb{E}[\hat{f}(\mathbf{x})])^2 - 2f\,\mathbb{E}[\hat{f}(\mathbf{x})] \\
&= Var[y] + Var[\hat{f}(\mathbf{x})] + \left(f^2 - 2f\,\mathbb{E}[\hat{f}(\mathbf{x})] + (\mathbb{E}[\hat{f}(\mathbf{x})])^2\right) \\
&= Var[y] + Var[\hat{f}(\mathbf{x})] + \left(f - \mathbb{E}[\hat{f}(\mathbf{x})]\right)^2 \\
&= \sigma_\epsilon^2 + Var[\hat{f}(\mathbf{x})] + Bias[\hat{f}(\mathbf{x})]^2
\end{aligned} \tag{3.3}$$

where we used $\mathbb{E}[\epsilon] = 0$, $\mathbb{E}[f] = f$, $\mathbb{E}[y] = f$ and $Var[x] = \mathbb{E}[x^2] - (\mathbb{E}[x])^2$. Above $\hat{f}(\mathbf{x})$ is an estimator that depends on the training set and is evaluated on the test input $\mathbf{x}$, while y is the vector of the test output. The expectation of the square error is taken with respect to both the training and the test data. Equation (3.3) shows that the prediction error is given by the sum of the noise variance, σ_ϵ^2, the variance of the predictive function $\hat{f}$, which is a measure of the variability of the predictions obtained by different training samples, and the squared bias, which is a measure of the error made by the mean predictor with respect to the target function. This decomposition of the squared prediction error highlights the so called *bias-variance dilemma*.

3.1.2 Model performance indicators

Many performance indicators are tipically used for the choice of the best model.
The most known is the Root Mean Square Error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\mathrm{y}_j - \hat{\mathrm{f}}_j\right)^2}$$

where y and $\hat{f}$ are the actual value and its forecast. RMSE derives from the L_2 loss function in fact it is the squared root of its expected value. Consequently, RMSE is sensitive to outliers because each error is weighted according to its squared value.
In the forecasting of energy time-series two widely used performance indicators are the Mean Absolute Error (MAE)

$$\mathrm{MAE} = \frac{1}{N}\sum_{j=1}^{N}\left|\mathrm{y}_j - \hat{\mathrm{f}}_j\right|$$

and the Mean Absolute Percentage Error (MAPE)

$$\mathrm{MAPE} = \frac{100}{N}\sum_{j=1}^{N}\frac{\left|\mathrm{y}_j - \hat{\mathrm{f}}_j\right|}{\mathrm{y}_j}$$

MAE is preferred over MAPE when the time series exhibits a non-stationary behavior or when the user is more interested to the absolute level of the error rather than its percentage value.
On the other hand, MAPE is preferred when the series is stationary or when different models are to be compared on several time series, with possibly different scales.

3.1.3 Backtesting and benchmarking

Probably, the most relevant parts in the development of a forecasting model are those devoted to backtesting and benchmarking.
The backtesting is the evaluation of model performances making predictions as of a past date, or based on a sample of simulated data. The dimensions of train and test sets for backtesting are crucial to obtain correct informations. If properly performed, the backtesting provides the error measure, by the chosen performance indicator, that could

be expected by the developed model on future data, under the fundamental hypotesis that future observations belong to the same distribution as the past ones.
The benchmarking, or model benchmarking, is the comparison of the performances of two or more models. The models in the comparison could range from the trivial to the most complex ones. This procedure highlights which is the best performer and sets the level for next improvements. The model benchmarking does not only select the most accurate model, but opens the way to model aggregation when two or more models, based on alternative methodologies, yield errors that, though comparable in size are not too much correlated.

3.1.4 Box-Jenkins models

Time series forecasting has been historically approached by Box-Jenkins models [15], which are methods based on the assumption that the time series is a realization of a stochastic process. These models rely on the analysis of the correlation of the time series assuming that the underlying stochastic process can be approximated by an ARMA (Autoregressive Moving Average) model

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) x_t = \mu + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \epsilon_t \tag{3.4}$$

in case of the time series is stationary, otherwise by an ARIMA (Autoregressive Integrated Moving Average) model

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) (1-L)^d x_t = \mu + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \epsilon_t. \tag{3.5}$$

In the equations 3.4 and 3.5, x_t is the time series data at time t, ϵ_t is the error term at time t, L is the lag operator, ϕ are the parameters of the autoregressive part, θ are the parameters of the moving average part, μ is the mean, p, d, q are the hyperparameters.
The SARIMA (Seasonal Autoregressive Integrated Moving Average) is one of the time series or Box-Jenkins models. In particular, it extends ARIMA taking into account the seasonal component of the series. In practice the analytic form of ARIMA(p,d,q) changes in SARIMA(p,d,q)(P,D,Q)$_s$

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right) \left(1 - \sum_{j=1}^{P} \Phi_j L^{j \times s}\right) (1-L)^d (1-L^s)^D x_t = \mu + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \left(1 + \sum_{j=1}^{Q} \Theta_j L^{j \times s}\right) \epsilon_t$$

where:

- Φ are the parameters of the seasonal autoregressive part
- Θ are the parameters of the seasonal moving average part
- P, D, Q, s are the hyperparameters of the seasonal part

This model could be further extended taking into account also exogenous variables, in which case the Box-Jenkins model is called SARIMAX.

3.2 Focus on considered models

In this section we will review the models developed and tested in the thesis.
Based on the availability of n data pairs $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$, known as the training data, a prediction rule $f(\cdot)$ is searched for with the objective of using $f(\mathbf{x}_*)$ as prediction of y_*, where $(\mathbf{x}_*, y_*)$ is any novel input-output pair. In this context, $\mathbf{x}_i \in \mathbb{R}^p$, $p < n$, is a vector whose entries are given by the p features associated with the target y_i.
In the following, with reference to the training data, $\mathbf{y} = y_i \in \mathbb{R}^n$ will denote the vector of the targets and $\mathbf{X} = \{x_{ij}\} \in \mathbb{R}^{n \times p}$ will denote the matrix of the training input data, where x_{ij} is the j-th feature of the i-th training pair $(\mathbf{x}_i, y_i)$.

3.2.1 Linear regression and his regularized extension

Ridge regression [16], LASSO [17] and elastic net [18] are methods to identify the parameters β_j of the linear-in-parameter predictor:

$$f(\mathbf{x}) = \sum_{j=1}^{p} x_j \beta_j = \mathbf{x}^T \boldsymbol{\beta}, \quad \boldsymbol{\beta} \in \mathbb{R}^p \tag{3.6}$$

where $\mathbf{x}^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$. Accordingly, the vector of the predicted training targets is

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x_1}) & f(\mathbf{x_2}) & \dots & f(\mathbf{x_n}) \end{bmatrix}^T = \mathbf{X}\boldsymbol{\beta} \tag{3.7}$$

To prevent overfitting and improve generalization capabilities, in all the three methods the loss function includes a penalty on the magnitude of $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}^{\text{ridge}} := \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|^2 \tag{3.8}$$

$$\boldsymbol{\beta}^{\text{LASSO}} := \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \sum_{i=1}^{p} |\beta_i| \tag{3.9}$$

$$\boldsymbol{\beta}^{\text{elastic net}} := \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \left(\alpha \|\boldsymbol{\beta}\|^2 + (1 - \alpha) \sum_{i=1}^{p} |\beta_i| \right) \tag{3.10}$$

The three methods share the same standard quadratic loss

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \sum_{i=1}^{n} L(y_i, f(\mathbf{x_i})), \quad L(y, \hat{y}) := (y - \hat{y})^2$$

but use different penalties that result in specific shrinking patterns.

Ridge regression

In ridge regression [19], the quadratic penalty shrinks parameters toward the origin.
This model could also be seen such as a simple linear model where the least squares loss is complemented with a constraint on the parameters sum of squares (Fig. 3.2)

$$\begin{cases} \beta^{ridge} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \\ \sum_{j=1}^{p} \beta_j^2 \leq t \end{cases}$$

Obviously, the hyperparameters λ and t are univocally interlinked.

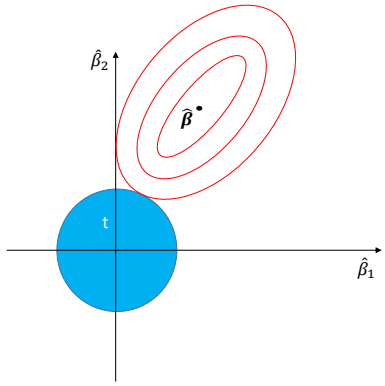


Fig. 3.2: Two dimensional example of ridge regression: the constraint on the parameters sum of squares is represented by the circular region. Its radius is a hyperparameter to be tuned.

Assuming that $\mathbf{X}$ is full rank, the solution of (3.8) is

$$\boldsymbol{\beta}^{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.11}$$

that highlights the shrinking effect with respect to the standard least squares estimator $\boldsymbol{\beta}^{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$.
Since the parameters are obtained in closed form (3.11), the ridge regression model is completely specified by the choice of λ, that can be calibrated following different approaches [19].
A normalized assessment of the amount of regularization associated with a given λ is provided by the so-called effective degrees of freedom

$$\text{df}(\lambda) = \text{tr}\left(\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\right)$$

In fact, $\text{df}(\lambda)$ ranges from p to 0 as λ goes from 0 to infinity [19].

LASSO

In the LASSO the penalty on the sum of absolute values has the effect of zeroing the least relevant parameters, thus enforcing some degree of sparsity.

This can also be seen as a linear model whose parameters are subject to a constraint on the sum of their absolute values:

$$\begin{cases} \beta^{LASSO} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j) \\ \sum_{j=1}^{p} |\beta|_j \leq t \end{cases}$$

where the hyperparameter t is directly linked to λ. Differently from the *Ridge* regression, there is not a closed form solution for the parameters. It is worth remarking that, depending on the choice of λ, the parameters may also shrink to zero as shown in Fig. 3.3. So the LASSO can be seen such as a continuous subset selection model.

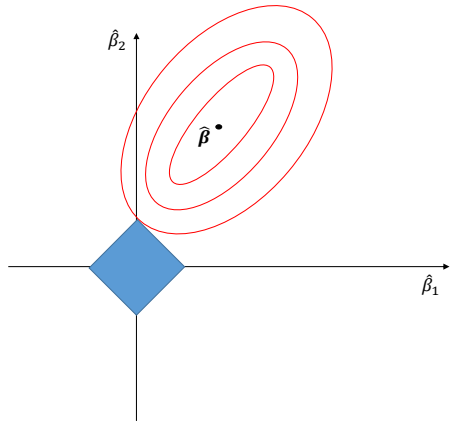


Fig. 3.3: A two dimensional representation of the LASSO. The graph shows that the LASSO has the potential to zero the value of some parameters.

The regularization hyperparameter λ (or t) gives higher generalization capacity to the model limiting the risk of overfitting. A common way to calibrate this hyperparameter is by cross validation.

Elastic Net

Another linear model is the *elastic net* which is a compromise between LASSO and ridge. The regularization term is given by a combination of the L_2 ridge penalty and the L_1 lasso penalty

$$\lambda \sum_{j=1}^{p} \left(\alpha\beta_j^2 + (1-\alpha)|\beta|_j\right) \tag{3.12}$$

where the additional parameter α is needed for weighting the LASSO and ridge components. In Fig. 3.4 is represented the elastic net threshold t in case of two features.

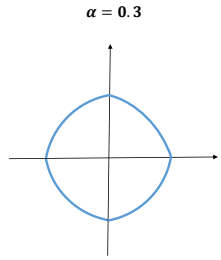


Fig. 3.4: A two factor representation of the elastic net threshold parameters. The second hyperparameter α is put equal to 0.3.

The regularization hyperparameters λ and α can be calibrated by cross validation.

3.2.2 Support vector machine

An alternative method to estimate f is by means of support vector regression (SVR), see e.g. [20]. In this case, in place of the quadratic loss, an ϵ-insensitive loss function is used:

$$L_\epsilon(y, \hat{y}) := \begin{cases} 0, & |y - \hat{y}| < \epsilon \\ |y - \hat{y}| - \epsilon, & \text{otherwise} \end{cases} \tag{3.13}$$

Moreover, the assumption is made that $f \in \mathcal{H}$, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) [21], whose reproducing kernel is denoted by $\kappa(\cdot, \cdot)$. Under this assumption,

$$\|f\|_\kappa = \sum_{j=1}^{\infty} \sum_{i=1}^{\infty} \alpha_j \alpha_i \kappa(x_j, x_i)$$

where α_i are such that

$$f(\mathbf{x}) = \sum_{j=1}^{\infty} \alpha_j \kappa(\mathbf{x}, \mathbf{x_j})$$

The SVR estimate is defined as

$$f^{\text{SVR}} := \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L_\epsilon(y_i, f(\mathbf{x_i})) + \frac{\lambda}{2} \|f\|_\kappa^2 \tag{3.14}$$

The hyperparameters are the real-valued constants λ and ϵ.

Although $f^{\mathrm{SVR}}(\mathbf{x})$ is a nonlinear function of $\mathbf{x}$, the *Representer Theorem* (see e.g. [22]) ensures that there exist coefficients c_i such that the predictor function can be written as a linear combination of kernel functions centered at $\mathbf{x_i}$

$$f^{\mathrm{SVR}}(\mathbf{x}) = \sum_{i=1}^{n} c_i \kappa(\mathbf{x_i}, \mathbf{x})$$

so that also the SVR predictor, though implementing a nonlinear function of the features, has a linear-in-parameter structure.

3.2.3 Gaussian processes

Let $\bar{\mathbf{y}} = \left[y_* \, \mathbf{y}^T \right]^T$, $\bar{\mathbf{x}} = \left[\mathbf{x}^{*T} \, \mathbf{x_1}^T \, \dots \, \mathbf{x_n}^T \right]^T$ and assume that, conditional on $\bar{\mathbf{x}}$, the vector $\bar{\mathbf{y}}$ is normally distributed as follows

$$\begin{aligned} \bar{\mathbf{y}} | \bar{\mathbf{x}} &\sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}(\bar{\mathbf{x}}) + \sigma^2 \mathbf{I_n}\right) \\ \left[\boldsymbol{\Sigma}(\bar{\mathbf{x}})\right]_{ij} &= \kappa(\bar{x}_i, \bar{x}_j) \end{aligned}$$

where the *kernel* $\kappa(\cdot, \cdot)$ is a suitable function whose choice reflects the available prior knowledge on the characteristics of the prediction rule. It is worth noting that the previous hypothesis is equivalent to assuming that

$$y_i = f(\mathbf{x_i}) + \epsilon_i, \quad i = 1, \dots, n$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are independent errors and $f(\cdot)$ is the realization a zero-mean continuous-time Gaussian Process (GP) with autocovariance $\kappa(\bar{\mathbf{x}}_\mathbf{i}, \bar{\mathbf{x}}_\mathbf{j})$[23, 24]. The estimation of a new target value y_* relies on the following property of normally distributed random vectors.

Lemma 3.1 (Distribution of jointly Gaussian variables). *Let z_* and $\mathbf{z}$ be jointly Gaussian random variables:*

$$\begin{bmatrix} z_* \\ \mathbf{z} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{z_* z_*} + \sigma^2 & \boldsymbol{\Sigma}_{z_* z} \\ \boldsymbol{\Sigma}_{z z_*} & \boldsymbol{\Sigma}_{zz} + \sigma^2 \mathbf{I_n} \end{bmatrix} \right)$$

Then, the posterior distribution of z_* conditional on z is:

$$z_* | \mathbf{z} \sim \mathcal{N}\left(\Sigma_{z_* z} \left(\boldsymbol{\Sigma}_{zz} + \sigma^2 \mathbf{I}_n \right)^{-1} \mathbf{z}, \; \Sigma_{z_* z_*} + \sigma^2 - \boldsymbol{\Sigma}_{z_* z} \left(\boldsymbol{\Sigma}_{zz} + \sigma^2 \mathbf{I}_n \right)^{-1} \boldsymbol{\Sigma}_{z z_*} \right)$$

In view of the previous lemma, it is possible to use the posterior expectation as prediction rule.

$$\begin{aligned} f(\mathbf{x}_*) &= \mathbb{E}\left[y_* | \mathbf{x}_*, \mathbf{y}, \mathbf{x} \right] = \sum_{i=1}^{n} c_i \kappa(\mathbf{x}_*, \bar{\mathbf{x}}_\mathbf{i}) \\ \boldsymbol{c} &= \left(\boldsymbol{\Sigma}(\mathbf{x}) + \sigma^2 \mathbf{I_n} \right)^{-1} \mathbf{y} \end{aligned}$$

The main distinctive feature of GP models is the learning process, which aims directly at obtaining the predictive function rather than inferring its parameters.

A zero-mean GP is completely defined by its covariance function $\kappa(\mathbf{x_i}, \mathbf{x_j})$, also called kernel. When it is a function of the distance $r = \|\mathbf{x_i} - \mathbf{x_j}\|$ between x_i and x_j, i.e. $\kappa(\mathbf{x_i}, \mathbf{x_j}) = \kappa(r)$, the kernel is said to be stationary and isotropic. Within this class, a popular and flexible choice is the family of Matérn kernels, defined by:

$$\kappa_{\text{Matern}}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)}\Big(\frac{\sqrt{2\nu}r}{l}\Big)^{\nu} K_{\nu}\Big(\frac{\sqrt{2\nu}r}{l}\Big)$$

where ν and l are hyperparameters to be tuned and K_ν is a modified Bessel function [25]. The parameter l defines the characteristic length-scale of the process, whereas ν defines the specific covariance function in the Matérn class. If ν tends towards infinity, the Matérn formula reduces to the widely used squared exponential function

$$\kappa_{\text{se}}(r) = \exp\Big(-\frac{r^2}{2l^2}\Big)$$

while if $\nu = 1/2$ it becomes an exponential function

$$\kappa_{\text{exp}}(r) = \exp\Big(-\frac{r}{l}\Big)$$

Different approaches are possible in order to tune the hyperparameters ν, λ, and σ^2. According to an empirical Bayes, the hyperparameter vector $\boldsymbol{\eta}$ is chosen as the maximizer of the marginal likelihood $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\eta})$.

3.2.4 Torus model

The torus model [26] is a linear model based on sinusoidal functions, originally developed to predict power load. Herein, its short-term version is adapted to forecast both gas and power demand series.
Following [26], a logarithmic transformation of the demand series D is performed in order to mitigate the effect of its skewness. The long-term model is

$$\ln \hat{\mathrm{D}}^{long}(t) = L(t) + F(t) + \sum_i H_i(t)$$

where the forecast is given by the sum of three elements: the trend or level L, the potential F, which accounts for seasonality, and the effect of holidays $\sum_i H_i$.
The potential F is modelled by a linear combination of sinusoidal functions:

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

where the functions h_i are given by the product of the j-th element in $\mathcal{D}$ with the k-th element in $\mathcal{W}$, for suitable j and k, and

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

The frequencies of the sinusoidal functions are $\Psi = \frac{2\pi}{365.25}$ and $\Omega = \frac{2\pi}{7}$. The number of harmonics, respectively N_w for 7-day and N_d for 365.25-day periodicity, are hyperparameters of the model.
We add to the original model of the potential F the linear dependency on temperature, expressed in $\mathrm{HDD}(t)$, and its daily difference $\mathrm{HDD}(t) - \mathrm{HDD}(t-1)$, by including these two features in the set of regressors. The terms related to trend and holidays are kept as presented in [26].
Finally, to get a short-term predictor, we correct the long-term model with the consumption of the previous day :

$$\hat{\mathrm{D}}(t) = \hat{\mathrm{D}}^{long}(t)\frac{\mathrm{D}(t-1)}{\hat{\mathrm{D}}^{long}(t-1)}.$$

The number of harmonics N_w and N_d were tuned by minimizing the Akaike information criterion (AIC).

3.2.5 K-nearest neighbor

K-Nearest neighbours (KNN) relies on the distance between samples in the feature space: given a test sample $\mathbf{x}_*$, the prediction of y_* is computed by averaging K training samples $y_i, i \in \mathcal{C}$, where $\mathcal{C}$ denotes the set identified by the K feature vectors $\mathbf{x_i}$ that are the closest to $\mathbf{x}_*$, according to some distance measure, e.g the Euclidean norm that was adopted herein.
In order to specify a KNN estimator, one has to choose the distance metric, e.g. Euclidean, Minkowsky, Manhattan, etc, and the type of weighted average, e.g. uniform or inverse distance, and to calibrate one hyperparameter, viz the number K of neighbours. Too small values of K lead to overfitting to the training data, while including too many neighbours reduces the variance at the cost of jeopardizing model flexibility.

3.2.6 Random forest

The Random forest method (see e.g. [24]) is based on so-called Classification and Regression Trees (CART) [27]. CARTs perform a recursive feature-wise partitioning of the input space and fit local linear regressions in each region of the final partition. CARTs are known to be unstable and prone to overfitting. In order to overcome these limitations, random forest models grow multiple CARTs, resorting to so-called data and feature bagging. Bagging or bootstrap aggregating is a random selection of a subset of a dataset that is repeated multiple times (with replacement). Models are then trained on each selected subset. By applying bagging to both data and features, each tree gets trained on different samples and feature sets. Forecasts performed by all models are then averaged to get the final prediction, leading to a more stable model.

3.2.7 Artificial Neural Networks

Artificial Neural Networks (ANN) are complex non-linear models, capable of capturing non-linear patterns and relations. A comprehensive explanation of their structure and the most common training algorithms can be found in [28].

There are three principal categories of ANN: the feedforward neural networks (FNN), the convolutional neural networks (CNN) and the recurrent neural networks (RNN). Historically, FNN were the first type of ANN: they introduce a non-linearity in the parameters by the application of a non-linear function, the activation function, to the typical linear combination of the feature maps (STEP 1). Moreover, they enhance the non-linearity by a multiple repetition of this operation followed by a second application of the activation function on the linear combination of the obtained results (STEP 2). The depth of the ANN grows increasing the number of repetitions of STEP 1 and STEP 2. Another characteristics of FNN is that each operation has an effect on the final results. In view of this, they are also called fully connected neural networks.
On the other hand, CNN are a type of neural networks where matrix multiplication is substituted by the convolution operation, which aims at finding the similarity between signals so as to identify patterns in the analyzed data. These neural networks are typically used for image identification because of their feature, appropriate for data with known grid-like topology. The new significant concepts introduced by CNN are the sparsity in the weights and the parameter sharing. Both these innovations help reducing the overparametrization and the consequent overfitting.
RNN are the third class of neural networks, specially suited to data with one-dimension known topology, such as for example time series. RNN recover the matrix multiplication of the FNN and have the important characteristics of parameter sharing just like CNN. The transition function between the different elements of the sequence, as well as its parameters, is shared along all the sequence, enhancing the generalization power of the model. The problem of the standard RNN is that they suffer from the so called vanishing gradient: the impact on the output of each input unit decays or increases exponentially step by step along the sequence of data, due to the multiple repetition of the gradient operation in the training optimization algorithm. Many architectures have been introduced to address this problem and, probably, one of the best solution as of today is given by the Long Short-Term Memory (LSTM) model [29],[30],[31],[32]. Another alternative RNN architecture, widely discussed in literature, [33], [34], is the Gated Recurrent Unit (GRU). In this thesis, we focus on two typologies of ANN: the Multi-Layer Perceptron (MLP), or feedforward neural network, and the RNN. For what concern the RNN, three different architectures will be explored and compared, the plain RNN, the LSTM and the GRU.

MLP

The MLP can be easily described as an extension of the linear model (eq. (3.6)) by three passages: first, in eq. (3.6) substitute the vector $\boldsymbol{\beta}$ of parameters with a matrix $\mathbf{W}$ of them; second, apply a non-linear function f, the so called activation function, to the obtained vector; third, compute the linear combination of the obtained results by another vector of parameters $\boldsymbol{\beta}$ in order to obtain the final scalar result. This description, reported in eq. (3.15),

$$y = f\left(\mathbf{x}^T\mathbf{W}\right)\boldsymbol{\beta} \tag{3.15}$$

regards the most simple MLP, where there is an input vector, a single layer, corresponding to the matrix $\mathbf{W}$ of parameters, and a scalar output.

The simple example of MLP with one layer is represented in fig. 3.5. In this case the vector of inputs is composed by four elements, the single layer has five neurons and finally there is a single output.
From this description, the extension to MLP with more than one layer is direct, being just a repetition of the steps yet described. The expression for a MLP with three layers is

$$y = f_3\left(f_2\left(f_1\left(\mathbf{x}^T\mathbf{W_1}\right)\mathbf{W_2}\right)\mathbf{W_3}\right)\boldsymbol{\beta}. \tag{3.16}$$

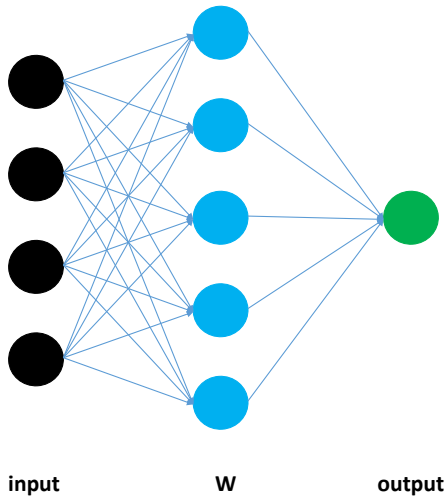


Fig. 3.5: A simple example of MLP with an input layer of 4 elements and one hidden layer with 5 neurons.

The MLP with three layers, that will be used to forecast the Italian gas and power demand, is represented in fig. 3.6. For clarity, the input vector with 22 elements is not included in the figure.

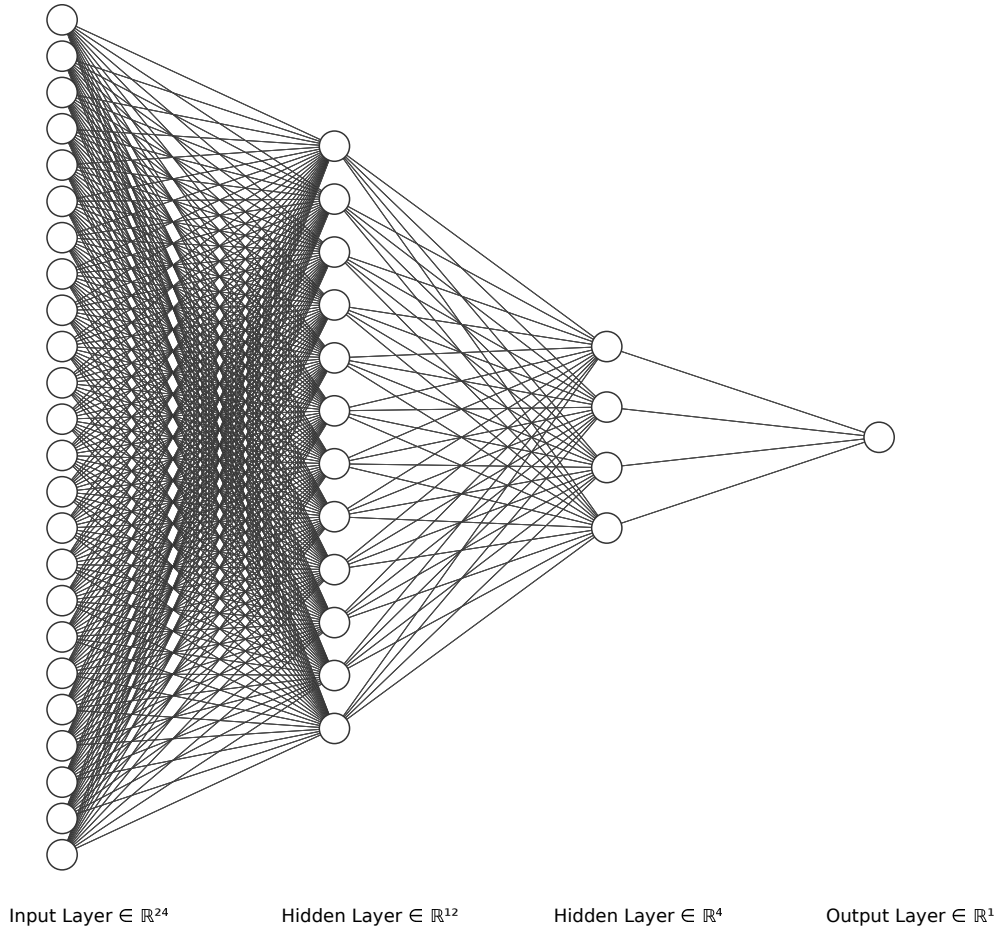


Fig. 3.6: The MLP implemented in this thesis with 22 input features, not displayed in the figure, three dense layers of 24, 12 and 4 neurons and an output neuron.

The Rectified Linear Unit (ReLu) activation function and the Mean Squared Error (MSE) loss function were adopted. Training was performed by means of gradient descent as implemented in the Adaptive Moment Estimation (ADAM) algorithm [35]. The tuned hyperparameters include the number of neurons in each layer, the parameters entering the definition of the activation functions, and optimization parameters such as number of epochs (the number of times that the learning algorithm works through the entire training set), batch size (the number of samples used for the updating of the model parameters), and learning rate (the step size in the gradient descent).

RNN

Differently from the FNN, the RNN are developed exactly for processing sequences of data and, based on a set of inputs, predict the next value of the sequence. The typical

RNN architecture (fig. 3.7) has an input layer, an hidden layer, composed by recurrent units, and, finally, an output layer. In order to fully understand this architecture, it is frequent to display the unfolded graph of it as well (fig. 3.7), which can also highlight the fundamental role of the parameter sharing at the base of the RNN.

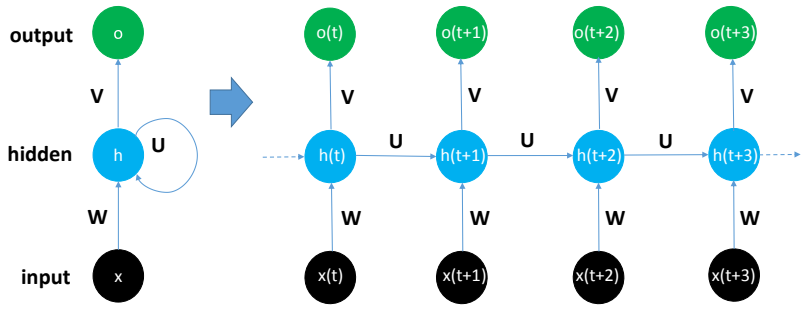


Fig. 3.7: The RNN architecture with its unfolded representation.

The hidden layer is represented by the state $\mathbf{h}$ of the system through the time t and $\mathbf{h}^{(t)}$ is given by

$$\mathbf{h}^{(t)} = f\left(\mathbf{h}^{(t-1)}, \mathbf{x}^t; \boldsymbol{\theta}\right) \tag{3.17}$$

where, f is the non-linear function $tanh$, $\mathbf{h}^{(t-1)}$ is the state vector at the previous time $t-1$, $\mathbf{x}^t$ is the vector of inputs at time t and $\boldsymbol{\theta}$ is the general vector of parameters which considers both the set of parameters $\mathbf{W}$ and $\mathbf{U}$ reported in fig. 3.7. This architecture lets the information flow through the time and, theoretically, gives the possibility to store it. The two matrices of parameters have dimension respectively $d \times n$ and $d \times d$, where n is the number of features, whereas d is the number of units in the hidden layer $\mathbf{h}$. The number of times considered, for which the parameters are shared, is usually called time window, or simply window. The total number of parameters involved in the simple RNN architecture is given by $d(n+d+1)$, where it is considered also the bias term.
On top of the hidden layer there is the output layer where the output is obtained by

$$\hat{\mathbf{y}}^{(t)} = \mathbf{o}^{(t)} = g\left(\mathbf{V}\mathbf{h}^{(t)}\right) \tag{3.18}$$

where, g is a generic non-linear function (e.g. sigmoid).
The plain RNN, just described, suffers from the vanishing or exploding gradient, [36], caused by the repeated product of real numbers (the gradients).

LSTM was introduced exactly to overcome this problem. In this architecture the cell state is protected by three different structures, called gates, which determine the flow of information in the cell so as to control and prevent the vanishing or exploding gradient. The LSTM cell, represented in fig. 3.8,

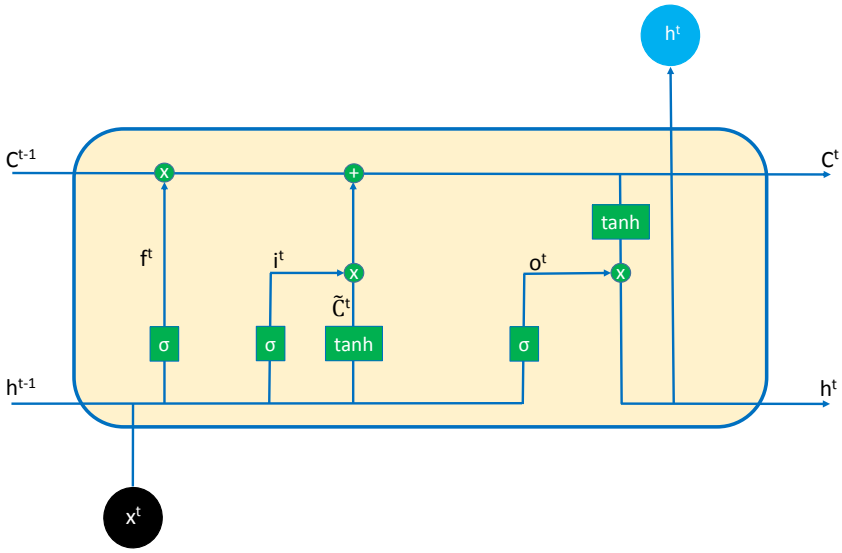


Fig. 3.8: The LSTM cell.

is clearly explained by its equations:

$$f^t = \sigma\left(\tilde{W}_f \cdot [h^{t-1}, x^t] + b_f\right) \tag{3.19}$$

$$i^t = \sigma\left(\tilde{W}_i \cdot [h^{t-1}, x^t] + b_i\right) \tag{3.20}$$

$$\tilde{C}^t = \tanh\left(\tilde{W}_C \cdot [h^{t-1}, x^t] + b_C\right) \tag{3.21}$$

$$C^t = f^t \odot C^{t-1} + i^t \odot \tilde{C}^t \tag{3.22}$$

$$o^t = \sigma\left(\tilde{W}_o \cdot [h^{t-1}, x^t] + b_0\right) \tag{3.23}$$

$$h^t = o^t \odot \tanh(C^t) \tag{3.24}$$

where, the equations of the three gates are eq. (3.19) for the *forget gate*, eq. (3.20) for the *input gate* and eq. (3.23) for the *output gate*. Equation (3.21) yields the component $\tilde{C}^t$ which represents the new information ready to update the memory cell C^t. Equation (3.22) returns exactly the updated value of the memory cell, where the value of the forget gate f^t represents the weight of the memory cell at the previous time, whereas the input gate

i^t controls the new information to be added to the memory cell. The third gate, the *output gate*, obtained with eq. (3.23), controls the output, multiplying the updated memory cell C^t, once it has been transformed by tanh so that it assumes values between -1 and 1 (eq. (3.24)). In all these equations the matrices of parameters $\tilde{W}_x$, with $x = f, i, o$ have dimensions $d \times n + d$, because $\tilde{W}_x = [U_x, W_x]$, and U_x have dimension $d \times d$, whereas W_x have dimension $d \times n$, with n number of features and d number of memory cells. The vector b_x, of dimension $d \times 1$, represents the constant term of each cell.
Thanks to the parameter sharing along the time, their number is reduced with respect to the MLP. The useful formula to determine the number of parameters of a LSTM cell is

$$\# \; of \; parameters = 4d(n + d + 1).$$

An alternative architecture to the LSTM, in order to overcome the vanishing gradient problem, is represented by the GRU. This architecture, introduced in 2014 ([33]), is similar to the LSTM one but has less parameters because it doesn't have the output gate.

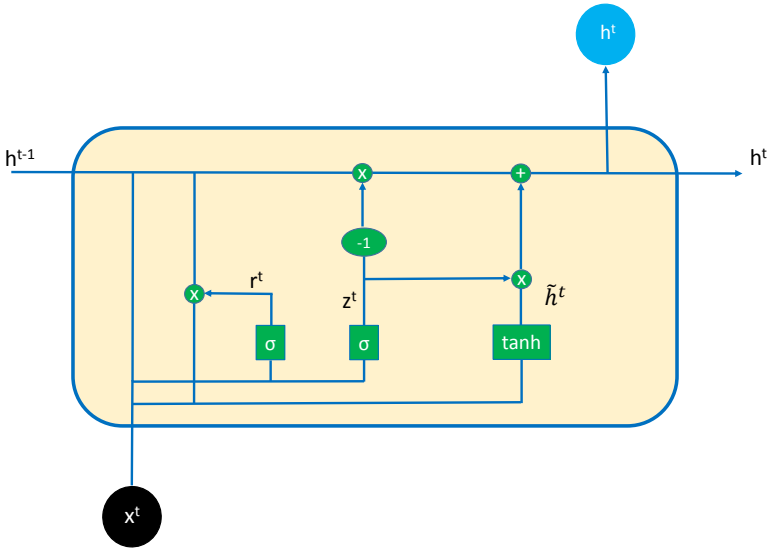


Fig. 3.9: The GRU cell.

The GRU architecture, represented in fig. 3.9, is completely described by its equations

$$
\begin{aligned}
z^t &= \sigma\left(\tilde{W}_z \cdot [h^{t-1}, x^t] + b_z\right) \\
r^t &= \sigma\left(\tilde{W}_r \cdot [h^{t-1}, x^t] + b_r\right) \\
\tilde{h}^t &= \tanh\left(\tilde{W}_h \cdot [r^t \odot h^{t-1}, x^t] + b_h\right) \\
h^t &= (1 - z^t) \odot h^{t-1} + z^t \odot \tilde{h}^t
\end{aligned}
$$

where, z^t is the update gate vector and r^t is the reset gate vector.
For the GRU architecture, the number of parameters is given by $3d(n + d + 1)$.
In the following, chapter 6, three plain RNN architectures will be considered, a simple RNN, a LSTM and a GRU, where all the configurations of each architecture are fixed except for the number of units, or memory cells, in a single hidden layer and the window, represented by the time steps to consider for the parameter sharing in the calibration. Thus the units and the window will be the only hyperparameters to calibrate.

3.3 Principal component analysis (PCA)

PCA is a statistical method that is part of the unsupervised learning techniques. As for the other unsupervised learning methods whose objective is to find relations among the regressors, PCA, based on a set of p regressors, aims to reduce the dimension of the problem by a projection on a space, with dimension $k < p$. Given the matrix of regressors $\mathbf{X} \in \mathbb{R}^{n \times p}$, where n is the number of samples and p the number of regressors, the first step of PCA is to compute the eigenvectors $\mathbf{V}$ $[p \times p]$ of the regressor covariance matrix where sum of the corresponding eigenvalues represents the amount of explained variance. These eigenvectors are such that, once chosen the first k of them (columns of $\mathbf{V}$), they simultaneously maximize the variance of the projection $\mathbf{XV}$ $[n \times k]$ and minimize the reconstruction error between $\mathbf{X}$ and $\mathbf{X}_{new}$, where $\mathbf{X}_{new} = \mathbf{XVV}^T$ is the reconstruction of the projected data in the original p-dimensional space.
In this thesis, PCA will be used in order to reduce the computational time in the hourly demand forecasting (Italian and zonal) and evaluate its potential regularization effect that is the result of its reduction of the number of parameters.
In order to maintain and be able to use the hourly/hourly-zonal temperature based regressors in case of a reduced number of components, we apply the PCA transformation identified on the hourly/hourly-zonal electric demand $\mathrm{IED_h}/\mathrm{XED_h}$ to the hourly/hourly-zonal temperature.

3.4 Ensemble methods

Ensemble models can have better performance than base ones because they can take advantage from the non-complete correlation between the forecasts obtained by the different base models. On the other hand sometimes, instead of the simple average of the base model forecasts, is necessary to weight the contribution of the base models to the ensemble one as a consequence of its own performance. Therefore four alternative aggregation techniques were considered. Apart from the Simple average, the calibration of the

other three methods requires a specific *ensemble training dataset* not used for training the base models.

Simple average
The most trivial aggregation is the arithmetic average of the forecasts achieved by the base models. Given a test input $\mathbf{x}_*$, and M base forecasts $\hat{f}_i(\mathbf{x}_*)$, $i = 1, ..., M$, the ensemble forecast is

$$\hat{f}^{\mathrm{A}}(\mathbf{x}_*) = \frac{1}{M} \sum_{i=1}^{M} \hat{f}_i(\mathbf{x}_*)$$

Weighted average
A second option is the weighted average of base forecasts:

$$\begin{cases} \hat{f}^{\mathrm{LS}}(\mathbf{x}_*) = \sum_{i=1}^{M} w_i \hat{f}_i(\mathbf{x}_*) \\ w_i \geq 0 \\ \sum_{i=1}^{M} w_i = 1 \end{cases}$$

The weights w_i are obtained by minimizing the sum of squared residuals between the ensemble forecast and the target vector on the ensemble training dataset.

Subset average (brute force)
The third ensemble method computes the average of a suitable subset of predictors. The chosen subset is obtained by a brute force search within the set of all possible subsets, choosing the subset of predictors whose average minimises the MAE computed on the ensemble training dataset. In our case, excluding the complete subset made of all the nine predictors (already considered as simple average), and the nine base models, the number of candidate subsets is

$$\sum_{k=2}^{8} \binom{9}{k} = 501$$

This ensemble method will be referred to as Subset average (b.f.).

Subset average (correlation analysis)
A further simple ensemble model tested is based on an intuitive algorithm to choose the optimum subset of $\bar{M} < M$ base predictors in order to obtain the final prediction by simply averaging the base model forecasts. We chose $\bar{M} = 3$ as trade-off between a low error and a meaningful average. The algorithm is designed so as to keep the most uncorrelated base forecasts and those with the minimum Mean Absolute Percentage Error (MAPE) at the same time.

Algorithm 1 A subset selection of $\bar{M}$ models.

```
for model_ a in models do
    if (model_ a not already excluded) and (∃ more than M̄ models left) then
        for model_ b in (models in descending order according to correlation(model_ a, model_ b))
        do
            if model_ b not already excluded then
                exclude the model (a or b) with bigger MAPE
            end
        end
    end
end
```

This ensemble method will be referred to as Subset average (c.a.).

SVR aggregation

The fifth ensemble method, called SVR aggregation, trains a SVR model on the ensemble training dataset, using base forecasts as features.

4

Italian gas demand forecasting

In this chapter the problem of Italian gas demand forecasting is addressed and the forecasts are obtained by the use of the machine learning models and ensemble methods described in chapter 3. The chapter is diveded in two principal sections which refer respectively to as many papers produced.
The first section is devoted to the Italian residential gas demand (RGD) forecasting, highlighting the fundamental role of temperature forecast in the RGD forecasting. Here we tested five machine learning models between those introduced in chapter 3. In this context, a main novel contribution of this work is the development of a model describing the propagation of temperature errors to gas forecasting errors that is successfully validated on experimental data. Being able to predict the quantitative impact of temperature forecasts on gas forecasts could be useful in order to assess potential improvement margins associated with more sophisticated weather forecasts. On the Italian data, it is shown that temperature forecast errors account for some 18% of the mean square error of gas demand forecasts provided by ANN.
The second section extends the first one in two directions: the forecasted time series and the set of considered models. Here the Italian gas demand (GD) forecasting is obtained such as the sum of the forecasts of each segment: RGD, industrial gas demand (IGD) and thermoelectric gas demand (TGD). For the predictions, the performances of all the nine machine learning models and four of the five ensemble techniques described in chapter 3 were analyzed.

4.1 Short-Term forecasting of Italian residential gas demand

4.1.1 Introduction and literature review

Forecasting natural gas demand is a crucial task for energy companies for several reasons. First, it provides relevant information to reserve pipe capacity and plan stocks effectively. Furthermore, regulations impose the balance of the network by charging providers with a fee proportional to their unbalanced quantity. Finally, demand is a critical input to forecast gas price, which is, in turn, a driver for business decisions.
Two comprehensive reviews of the literature about gas demand forecasting are [2] and [3]. According to Sebalj et al. [2], papers can be classified along four dimensions. The

prediction horizon can range from hourly to yearly, the *reference area* from single nodes of the network to a whole country; adopted *models* include time series, mathematical and statistical approaches, neural networks, and others; input *features* can be demand history, temperature, calendar, and other minor ones.

Several studies focused on country- or regional-level daily forecasting. Mathematical and statistical models based on parametric non-linear functions were used in [37] to explain the factors which affect the demand. A different multi-factor approach was developed in [38] and a model based on the physical relation between gas demand and the temperature was presented in [39]. An adaptive network-based fuzzy inference system (ANFIS) was described in [40], where the authors showed better performances of their model concerning ANN and conventional time series methods. A statistical learning model, based on support vector machine (SVM), was developed in [41] for UK demand, and compared to ANN and an autoregressive moving average (ARMA) predictor. A hybrid model, exploiting many different techniques, such as wavelet transform, genetic algorithm, ANFIS, and ANN, was used in [42]. Neural networks were applied in [43, 44, 45, 46, 47] to perform hourly and daily forecasts on cities and regions. Moreover, [48] showed how ANNs, combined with Principal Components Correlation Analysis (PCCA), provide robust and precise forecasts on regional demand. Baldacci et al. [49] used nearest neighbors and local regression to forecast the gas demand of small villages. They also presented an investigation over the effects of temperature forecast errors, concluding that the influence on model accuracy is negligible.

Concerning long-term forecasting, [50] discussed gas demand in Bangladesh, showing how population growth and Gross Domestic Product (GDP) are essential drivers of the demand. Similar conclusions were achieved in [51], where a breeder model was proven superior to other approaches in forecasting Turkish demand.

The present work focuses on day-ahead forecasting of Residential Gas Demand (RGD) at country level. In particular, Italian RGD is used as a case study to try and fill two gaps in the existing literature, revealed by our review.

First, a comparison among five forecasting methods of different nature was carried out, two based on linear regression and three on machine learning techniques, with the aim of uncovering strengths and weaknesses of each one, paying particular attention to their accurate tuning. This involves a detailed discussion on the selection of the relevant covariates, among which a primary role is played by the weather temperature.

The second gap has to do with the influence of weather forecast errors on natural gas demand models. Despite being critical in industrial applications, previous works seldom specify if the predictors use forecasted or observed temperature, maybe due to the belief that temperature errors have negligible impact. In contrast, we assess the influence of weather forecasting errors, both theoretically and experimentally. A novel easy-to-compute bound is derived that predicts the best achievable RGD root mean square error (RMSE) as a function of the temperature RMSE. This bound is then validated on experimental data: Italian RGD forecasts are obtained using both observed and predicted temperatures, thus allowing for a quantitative assessment of accuracy degradation.

The work is organized as follows. In section 4.1.2, the problem is formulated and the available data are presented. In section 4.1.3, a statistical characterization of target and input variables are provided, discussing both preprocessing and feature selection. Section 4.1.4 describes models, including the training process and hyperparameter tuning. In section 4.1.5, we derive the performance limit, which is used as the ultimate benchmark

in section 4.1.6, where the results are presented and discussed. Finally, section 4.1.7 is devoted to some concluding remarks.

4.1.2 Problem Statement

In Italy, natural gas is the most common fuel for both power plants and domestic heating. Moreover, several industrial facilities burn gas for either heating or powering productive processes. According to SNAM Rete Gas [52], the Italian Transmission System Operator (TSO), in 2017 about 70.59 billions of cubic meters of natural gas were consumed, with an increase of 5.6% over the previous year. Overall, the increase in demand between 2015 and 2017 was 11%. Out of the total gas demand in 2017, 35.9% was due to thermoelectric power plants, 22.4% to industrial facilities, and 41.7% to residential users.
The task addressed in this work is the one-day-ahead forecasting of daily Italian Residential Gas Demand (RGD). RGD represents the main part of the overall Italian gas consumption, accounting for household usage for cooking, water heating, and, most importantly, environment heating.
The available dataset covers 11 years, from 2007 to 2017, and is made of 3 fields: date (t), forecasted average temperature ($\hat{T}$) and residential gas demand (RGD). Forecasted temperature is relative to the Northern regions of Italy. In the preliminary analysis, we also took into consideration a weighted average of the temperatures in different zones of Italy, but a weaker correlation with RGD was noticed. This is explained by the role of domestic heating in Northern Italy, where winters are colder than in other regions.
The profile of RGD from 2007 to 2018 is displayed in fig. 2.2.

4.1.3 Exploratory analysis and feature selection

4.1.3.1 Residential Gas Demand

As observed in section 2.3 RGD magnitude greatly oscillates with the season following the changes of temperature during the year.
The characteristics of RGD explained in section 2.3 are evident in fig. 4.1: a pronounced yearly periodicity, given by the significant link between RGD and temperature, and a strong weekly periodicity during the warm season when temperature is above 17-18 Celsius degrees so that its effect on RGD is negligible.

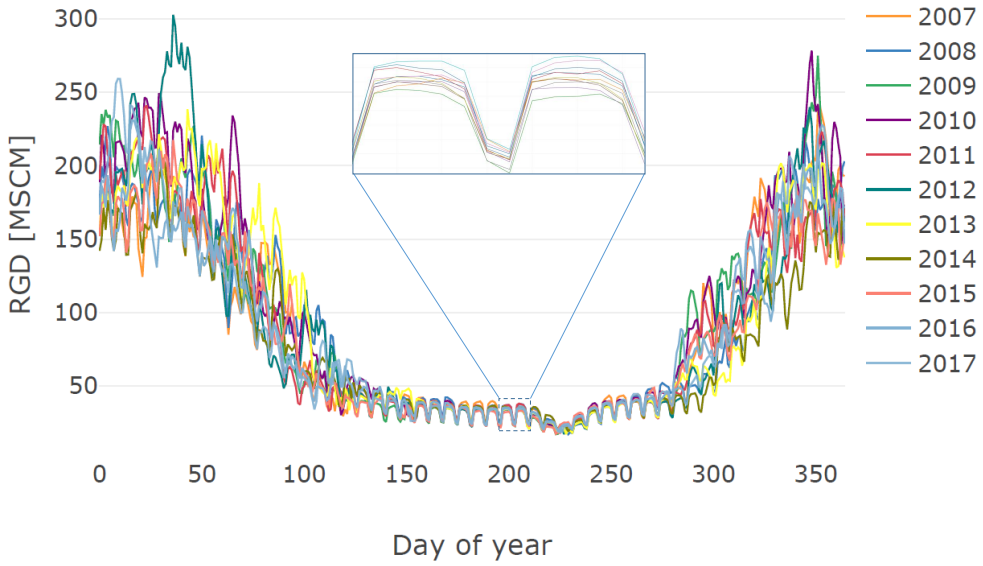


Fig. 4.1: Italian Residential Gas Demand (RGD): years 2007-2017. The time series are shifted to align weekdays: weekly periodicity is particularly visible in summer. The yearly seasonal variation is mostly explained by heating requirements. In the inset, two weeks of July's demand are zoomed.

As expected, the autocorrelation function, estimated on the whole dataset, exhibits a clear yearly seasonality and a much smaller weekly periodicity, see fig. 4.2.
Most of the spectral density, see fig. 4.3, is concentrated at period 365.25 days. A smaller yet relevant spike can be found at a period of 7 days, accounting for the weekly periodicity. In both cases, smaller peaks at lower periods are ascribable to harmonics.

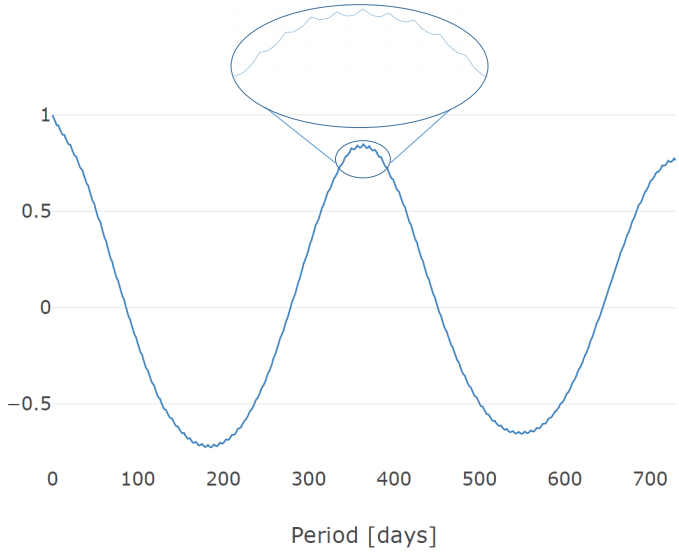


Fig. 4.2: RGD autocorrelation function estimated on 2007-2017 data. The 365-day yearly periodicity is evident. In the inset, weekly waves witness the presence of a 7-day periodicity of smaller amplitude.

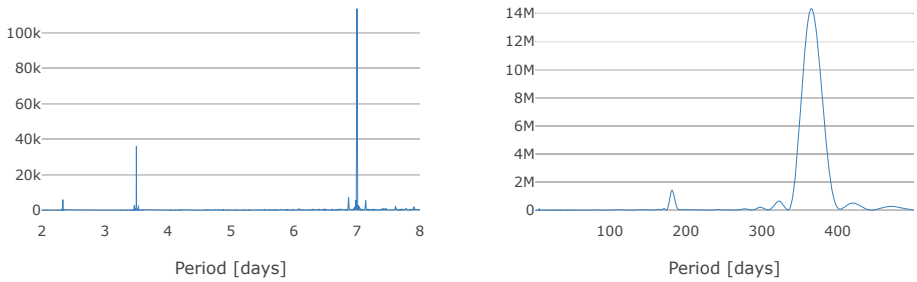
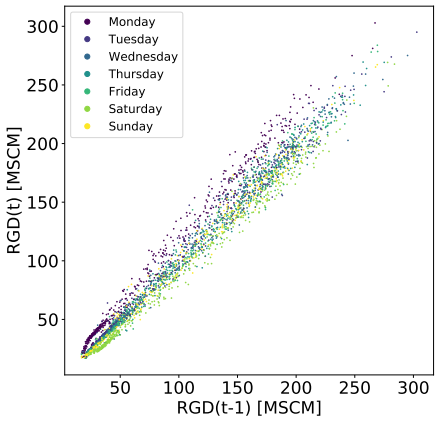


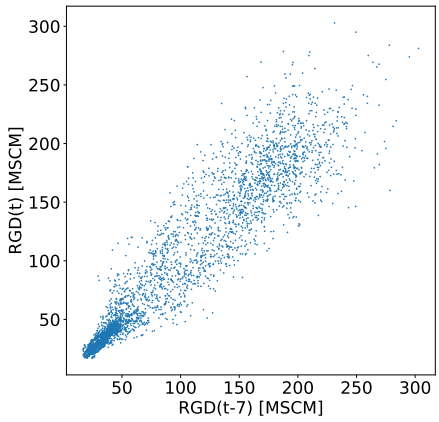
Fig. 4.3: RGD periodogram. Left panel: periods from 0 to 8 days; right panel: periods from 0 to 500 days. The yearly periodicity is highlighted by peaks at 365.25 days, while the weekly one by the smaller spike at a period of 7 days. Other notable values are caused by harmonics.

The autocorrelation of lag 1 can be assessed through the scatter plot in fig. 4.4a, where RGD at time t is plotted against RGD at time $t-1$. The correlation coefficient computed on the entire dataset is 0.988, and it increases to 0.995 if Saturdays and Mondays are discarded. This is an evidence of a different behavior between working days and weekends,

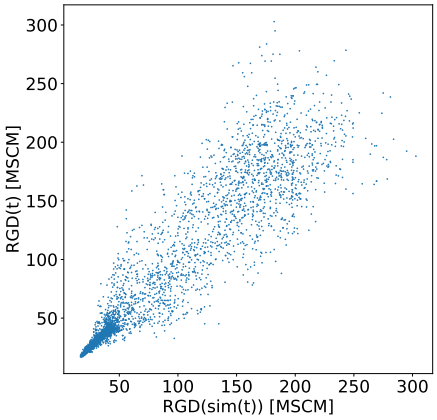
visually confirmed in the plot, where Monday's RGD (orange dots) stays in the upper part of the cloud whereas Saturday's RGD (green dots) lies in the lower part.
As for the lag-7 autocorrelation, in fig. 4.4b the scatter plot of RGD at times t and $t-7$ is displayed. The scatter plot in fig. 4.4b is narrower when the demand is low, that is during warm months, while it gets wider in winter when the demand is high. This is due to the variability of weather from one week to the next one.
In order to characterize the yearly seasonality, the relation between RGD at time t and RGD in the similar day was analysed (cf. Def. 2.1).
The relationship between RGD and RGD in the similar day is shown in fig. 4.4c: again, the correlation is higher when the demand is lower, due to the smaller influence of temperature.
It can also be of some interest to take into account the similar day of $t-1$. The scatter plot in fig. 4.4d shows that the difference $\mathrm{RGD}(t-1) - \mathrm{RGD}(\mathrm{sim}(t-1))$ is a good proxy to the difference $\mathrm{RGD}(t) - \mathrm{RGD}(\mathrm{sim}(t))$.
Due to these considerations, we use $\mathrm{RGD}(t-1)$, $\mathrm{RGD}(t-7)$, $\mathrm{RGD}(\mathrm{sim}(t))$, and $\mathrm{RGD}(\mathrm{sim}(t-1))$ as inputs to forecast RGD at time t.



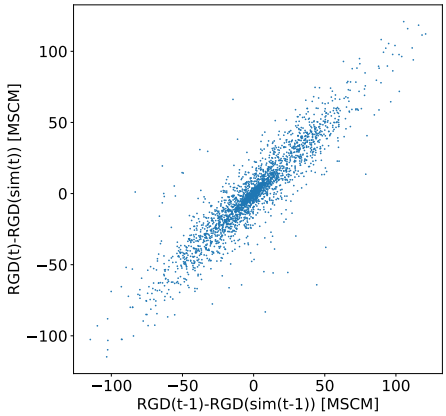
(a) RGD(t) vs RGD($t-1$)



(b) RGD(t) vs RGD($t-7$)



(c) RGD(t) vs RGD(sim(t))



(d) RGD(t) − RGD(sim(t)) vs RGD($t-1$) − RGD(sim($t-1$))

Fig. 4.4: Scatter plots between RGD and potential features to be used for its prediction.

4.1.3.2 Temperature

The RGD time series shows a strong relation with temperature, especially when, during the winter season, temperature falls below 18°C and household heating becomes relevant. As shown in the left panel of fig. 4.5, the relationship is piecewise linear: a line with negative slope below 18°C, followed by an approximately constant line above 18°C. In

order to transform the piecewise linear dependence into a linear one, it is useful to make reference to the so-called Heating Day Degrees (HDD):

Definition 4.1 (Heating Day Degrees (HDD)).

$$\mathrm{HDD}(T) = \max(18^\circ - T, 0) \tag{4.1}$$

In the right panel of fig. 4.5, the scatter plot of RGD vs HDD highlights an approximately linear relationship, with a positive correlation of 0.97. The correlation of HDD with RGD is even more evident when we look at the time series of RGD and HDD during 2017, see fig. 4.6.

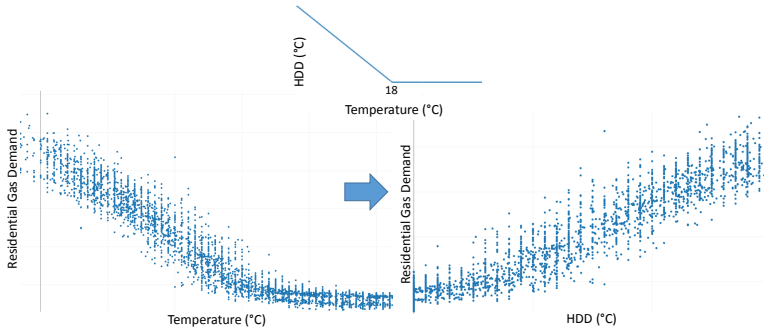


Fig. 4.5: Left panel: scatter plot of daily RGD vs average daily temperature. Right panel: scatter plot of daily RGD vs HDD. Inset: HDD as a function of the temperature.

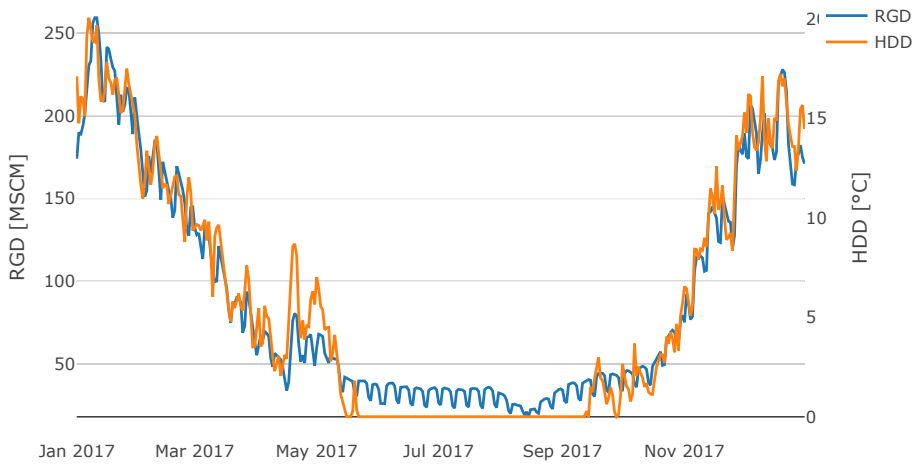


Fig. 4.6: Time series of RGD and HDD in 2017. The instantaneous correlation between the two series is apparent.

As shown in fig. 4.5, HDD are more correlated to gas demand than plain temperatures. Thus, $\mathrm{HDD}(\hat{T}(t))$ is considered as a feature, where $\hat{T}(t)$ denotes the one-day-ahead forecast of $T(t)$. As additional features also $\mathrm{HDD}(\hat{T}(t-1)), \mathrm{HDD}(\hat{T}(t-7)), \mathrm{HDD}(\hat{T}(\mathrm{sim}(t)))$ are included. For completeness, also $\hat{T}(t), \hat{T}(t-1)$, $\hat{T}(t-7)$ and $\hat{T}(\mathrm{sim}(t))$ are considered. The choice of using the forecasting temperatures at past times instead of the real ones is motivated by the need to replicate the relationship between tomorrow's gas demand and temperature forecast.

4.1.3.3 Calendar features

As shown in the previous paragraphs, weekdays and holidays have a great influence on RGD. To capture this phenomena, the following categorical calendar features are taken into account.

Weekday. In view of the weekly periodicity, the seven days of the week are taken as explanatory features. By resorting to the one-hot encoding method they are transformed in 7 dichotomic time series.

Holiday. A binary feature which takes value 1 in correspondence of holidays.

Day after holiday. A binary feature which takes value 1 the first working day after a holiday. A working day is a day different from Saturday and Sunday that is not a holiday.

Bridge holiday. A binary feature which takes value 1 on isolated working days, that is working days where both the day before and the day after are either Saturday, Sunday or a holiday.
All the features are summarized in table 4.1.

Feature	Reference time	Type
RGD	t-1	continuous
RGD	t-7	continuous
RGD	sim(t)	continuous
RGD	sim($t-1$)	continuous
Forecasted temperature	t	continuous
Forecasted temperature	t-1	continuous
Forecasted temperature	t-7	continuous
Forecasted temperature	sim(t)	continuous
Forecasted HDD	t	continuous
Forecasted HDD	t-1	continuous
Forecasted HDD	t-7	continuous
Forecasted HDD	sim(t)	continuous
Weekday	t	categorical
Holiday	t	binary
Day after holiday	t	binary
Bridge holiday	t	binary

Table 4.1: List of features

4.1.4 Predictive models and implementation notes

The classical methods used for time series forecasting are linear Box-Jenkins models such as SARIMA, where the forecast is based only on past values of the time series, and SARIMAX, that accounts also for exogenous variables. A major drawback of classical linear models is given by discontinuities due to holidays and the possible presence of other nonlinear phenomena. In order to overcome these difficulties, herein RGD forecasting is formulated as a statistical learning problem.
Based on the availability of n data pairs $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$, known as the training data, a prediction rule $f(\cdot)$ is designed with the objective of using $f(\mathbf{x}_*)$ as prediction of y_*, where $(\mathbf{x}_*, y_*)$ is any novel input-output pair. In this context, $\mathbf{x}_i \in \mathbb{R}^p$, $p < n$, is a vector whose entries are given by the p features associated with the target y_i.
Herein, the p features are the 22 covariates discussed in the previous section and shown in table 4.1. In the following, with reference to the training data, $\mathbf{y} = y_i \in \mathbb{R}^n$ will denote the vector of the targets and $\mathbf{X} = \{x_{ij}\} \in \mathbb{R}^{n \times p}$ will denote the matrix of the training input data, where x_{ij} is the j-th feature of the i-th training pair $(\mathbf{x}_i, y_i)$.
We implemented and tested the following models, described in chapter 3:

- ridge regression;

- torus model [26];
- Gaussian Process (GP);
- k-nearest neighbour (KNN);
- artificial neural network (ANN). In particular we tested the MLP architecture shown in fig. 3.6.

All the models, except the torus one, were implemented in Python, using scikit-learn and keras; automated hyperparameters tuning exploited the GridSearchCV function of scikit-learn. The torus model was implemented in MATLAB, as well as its hyperparameter tuning routine.

4.1.5 Effects of temperature forecast errors

As shown in section 4.1.3, the temperature is the most important exogenous variable. Obviously, the actual temperature cannot be used when forecasting future RGD: only a forecast is available, affected by a small yet non-negligible error, which inevitably impacts also the performance of gas demand forecast. The scope of this section is to assess the influence of the temperature error on the precision of the RGD forecast. For this purpose, we resort to an idealized error propagation model that, despite its simplicity, provides an accurate description of this effect, as confirmed by the subsequent experimental validation. Let RGD be a deterministic function g of the true temperature T and some other factors $\mathbf{x} = (x_1, x_2, ...)$: $\mathrm{RGD} = g(T, \mathbf{x})$. In view of the analysis and the charts presented in section 4.1.3, a first-level approximation of the relationship between RGD and T is a linear function of HDD, while the dependence on the other factors can be represented as an additive term $\bar{g}(\mathbf{x})$:

$$\mathrm{RGD} = g(T, \mathbf{x}) = \bar{g}(\mathbf{x}) + \alpha \mathrm{HDD}(T)$$

where α is the sensitivity of the gas demand to HDD. The formula is of general validity and applies to both regional and national gas markets. Indeed, α depends on the size of the considered market and can be estimated from historical data, e.g. those displayed in fig. 4.5.
Consider now the ideal case when α and also the function $\bar{g}$ are perfectly known, yet, only a forecast

$$\hat{T} = T + \epsilon$$

of the correct temperature T is available, where ϵ is a zero-mean error with variance σ_ϵ^2. The optimal forecast $\hat{\mathrm{RGD}}$, given $\hat{T}$, is therefore:

$$\hat{\mathrm{RGD}} = \bar{g}(\mathbf{x}) + \alpha \mathrm{HDD}(\hat{T})$$

In order to obtain the mean squared error of $\hat{\mathrm{RGD}}$, the conditional variance of $\hat{\mathrm{RGD}}$ is first computed:

$$\mathrm{Var}\left[\hat{\mathrm{RGD}} \mid T \geq 18°\right] = \mathrm{Var}\left[\bar{g}(\mathbf{x}) + \alpha \cdot 0\right] = 0$$
$$\mathrm{Var}\left[\hat{\mathrm{RGD}} \mid T < 18°\right] = \mathrm{Var}\left[\bar{g}(\mathbf{x}) + \alpha\left(18° - \hat{T}\right)\right] = \alpha^2 \mathrm{Var}\left[\epsilon\right] = \alpha^2 \sigma_\epsilon^2$$

Since $\mathbb{E}[\epsilon] = 0$, it follows that $\mathbb{E}[\hat{\mathrm{RGD}}] = \mathrm{RGD}$. Thus:

$$\begin{aligned}\mathbb{E}\left[\left(\hat{\mathrm{RGD}}-\mathrm{RGD}\right)^2\right] &= \mathbb{E}\left[\left(\hat{\mathrm{RGD}}-\mathrm{RGD}\right)^2 \mid T \geq 18^\circ\right] P\left(T \geq 18^\circ\right) + \\ &+ \mathbb{E}\left[\left(\hat{\mathrm{RGD}}-\mathrm{RGD}\right)^2 \mid T < 18^\circ\right] P\left(T < 18^\circ\right) = \\ &= 0 + \mathrm{Var}\left[\hat{\mathrm{RGD}} \mid T < 18^\circ\right] = \\ &= P\left(T < 18^\circ\right) \alpha^2 \sigma_\epsilon^2 \end{aligned} \tag{4.2}$$

This last equation provides an estimate of the mean squared error due to the temperature forecasting error. Since it has been derived under an ideal setting - α and $\bar{g}(\cdot)$ perfectly known, it provides a lower limit to the precision that can be achieved by the best possible forecaster.
The arguments entering the bound are easily obtainable as follows:

- Estimate $P\left(T < 18^\circ\right)$ by computing the ratio between the number of samples such that $T < 18^\circ$ and the total number of available data.
- Compute α through a least square fit of RGD vs T.
- Estimate σ_ϵ^2 as the sample variance of $\hat{T} - T$.

Considering the Italian RGD data, in the 3-year period 2015-2017, $P\left(T < 18^\circ\right)$ ranges from 54% to 67%, while σ_ϵ^2 ranges from 0.05 to 0.09, and α from 9.85 to 10.96. Considering altogether the years 2015-2017, we have $P\left(T < 18^\circ\right) = 63\%$, $\sigma_\epsilon^2 = 0.063$, $\alpha = 10.56$, corresponding to a best achievable Root Mean Squared Error

$$\mathrm{RMSE} = \sqrt{\mathbb{E}\left[\left(\hat{\mathrm{RGD}}-\mathrm{RGD}\right)^2\right]} = 10.56 \times \sqrt{0.63 \times 0.063} = 2.22 \ \ \mathrm{MSCM}$$

Finally, we consider the more realistic case in which the forecasting mean square error is different from zero even in absence of temperature errors, that is

$$\mathrm{Var}\left[\bar{g}\left(\mathbf{x}\right)\right] = \sigma_0^2 > 0$$

Then, under a statistical independence assumption, it is possible to obtain the forecasting RMSE as a function of σ_ϵ^2:

$$\mathrm{RMSE}(\sigma_\epsilon^2) = \sqrt{\mathrm{Var}\left[\hat{\mathrm{RGD}}\right]} = \sqrt{\sigma_0^2 + P\left(T < 18^\circ\right) \alpha^2 \sigma_\epsilon^2} \tag{4.3}$$

In fig. 4.7 this relationship is displayed assuming $P\left(T < 18^\circ\right) = 63\%$, $\sigma_\epsilon^2 = 0.063$, $\sigma_0^2 = 13.31$ (this last value is the test MSE achieved by the ANN forecaster trained with true temperature data instead of the forecasted ones, see Results). Notably, the sensitivity of the gas forecasting error tends to increase as the temperature forecast error grows. In particular, if the threshold is defined

$$\bar{\sigma}_\epsilon^2 = \frac{\sigma_0^2}{P\left(T < 18^\circ\right) \alpha^2}$$

the influence of temperature errors is negligible as far as $\sigma_\epsilon^2 \ll \bar{\sigma}_\epsilon^2$, while the temperature errors have a linear influence on the gas RMSE for $\sigma_\epsilon^2 \gg \bar{\sigma}_\epsilon^2$.

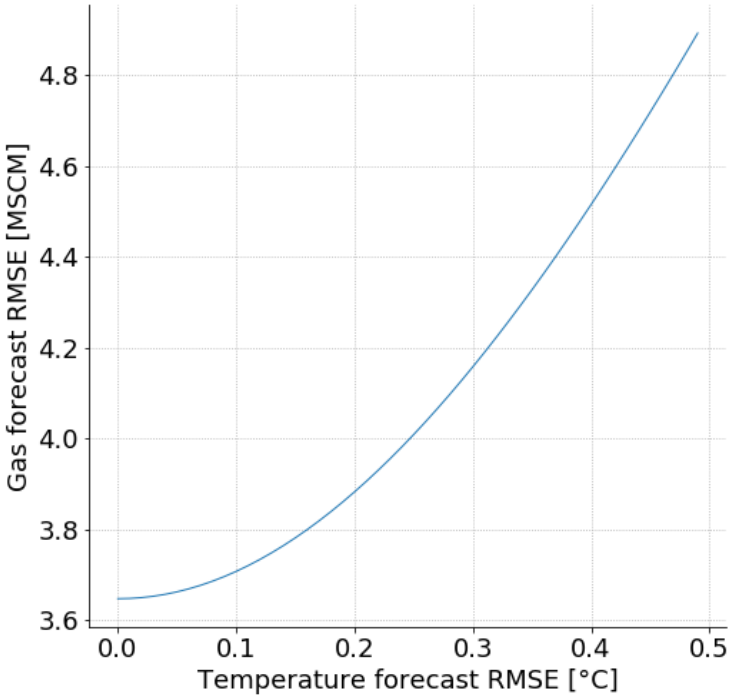


Fig. 4.7: The relation between the Gas forecast RMSE and the Temperature forecast RMSE.

4.1.6 Results

4.1.6.1 Data and performance indicators

As mentioned in section 4.1.2, available data range from 2007 to 2017. Three one-year-long test sets were defined, associated with the year 2015, 2016, and 2017. The corresponding training sets spanned from 2007 to the day before the start of the test set: 2007-2014, 2007-2015, 2007-2016. In the following, each training set is identified by the year of the corresponding test set, e.g., we will write "training set 2016" to indicate the second training set, spanning from 2007 to 2015.
On each test set, the performance of the five models was measured using the Mean Absolute Error (MAE).

$$\mathrm{MAE} = \frac{1}{N}\sum_{j=1}^{N}\left|\mathrm{RGD}_j - \hat{\mathrm{RGD}}_j\right|$$

MAE is preferred over MAPE due to the highly non-stationary behavior of RGD series. Using MAPE would attribute undue importance to errors during the summer period when

RGD is small, see fig. 4.1. Moreover, MAE is proportional to the monetary loss sustained by energy companies because of errors in nomination due to inaccurate forecasts.
Nevertheless, in order to allow a comparison with forecasting performances achieved in the UK market, we will also refer to the MAPE

$$\text{MAPE} = \frac{100}{N} \sum_{j=1}^{N} \frac{\left| \text{RGD}_j - \hat{\text{RGD}}_j \right|}{\text{RGD}_j}$$

The comparison between two different markets calls for the use of a relative metric. To avoid the confounding effect of small absolute errors that are amplified by MAPE during the Italian Summer, the comparison between Italy and UK was limited to the cold months, when gas demand is relatively high, see Section 4.1.6.3.
Finally, as the performance limit derived in section 4.1.5 poses a lower bound to the mean squared error, the Root Mean Square Error (RMSE) was also used as a comparison metric.

4.1.6.2 Hyperparameters

All five models include hyperparameters that were tuned by cross-validation.
For ridge regression, the regularization parameter λ was tuned by 5-fold cross-validation on an interval ranging from 10^{-4} to 10^{2} in logarithmic steps. In the training set 2015, line search selected $\lambda = 0.236$, corresponding to $\text{df}(0.236) = 20.94$, while in the other two sets, 2016 and 2017, cross-validation selected the minimum $\lambda = 10^{-4}$ with effective degrees of freedom $\text{df}(10^{-4}) = 20.99$ practically equal to the number of parameters. This means that regularization plays a very marginal role.
For KNN, the number of neighbors, in the interval $[1, 30]$, were optimized and also the weighting strategy, choosing between uniform and inverse of the distance. Seven neighbors were obtained for training set 2015 and 6 for the two remaining ones. In all the three cases, the "inverse of distance" weights were selected.
As for the Gaussian Process, the maximization of the marginal likelihood yielded $\nu = 1.5$, $l = 10$, and $\sigma^2 = 10$, with minimal variations among all training sets.
For the ANN models, a trial and error procedure led to architecture with an input layer of 24 neurons, two hidden layers of 12 and 4 neurons, and an output layer of a single neuron, as shown in fig. 3.6. By 5-fold cross-validation, a learning rate of 0.001, a number of epochs of 1000, and a batch size of 32 were obtained.
For what concerns the Torus model, the minimization of AIC led to the choice of $N_w = 3$ and $N_d = 1$ for all the training sets.

4.1.6.3 Prediction results

A first assessment of the performances of the adopted methods was carried out in terms of RMSE. In order to validate the formula that models the propagation of temperature errors (section 4.1.5), two sessions were performed. In the first one, the models were trained and tested using historical records of true temperatures, assuming that the one-day-ahead exact temperature is available as a feature. Then we used eq. (4.3) in order to predict how much the forecasting RMSE would increase in the more realistic scenario in which one-day-ahead temperature forecasts are employed in place of the true temperatures. In the

second session, the models were trained and tested using historical records of forecasted temperatures. In this way, it was possible to validate the error propagation model against the real errors.
The results of the first session are summarized in table 4.2. It can be seen that the smallest RMSE is obtained by GP and ANN, the latter being marginally better.

Year	2015	2016	2017	2015-2017
Ridge	4.24	4.11	4.12	4.16
GP	3.81	3.68	3.64	3.71
KNN	7.29	8.49	8.38	8.07
Torus	4.21	4.23	3.70	4.05
ANN	3.89	3.60	3.44	3.65

Table 4.2: Performance on test sets: yearly RMSE (MSCM) of the five forecasters trained and tested assuming that one-day-ahead true temperatures are available.

Obviously, in a real-world context, only temperature forecasts are available for the day ahead. In order to account for the performance degradation due to the use of forecasted temperatures, eq. (4.3) was used to predict the RMSE of the RGD forecast in correspondence of a temperature forecast variance $\sigma_\epsilon^2 = 0.063$, coinciding with that of our meteorological data. The results are summarized in table 4.3. In the first line, the theoretical performance limits computed according to eq. (4.2) are reported. These values were added to the RMSE's of table 4.2 to obtain predictions of RGD forecasting RMSE in a real-world situation in which one-day-ahead temperature forecasts are used.

Year	2015	2016	2017	2015-2017
Performance limit	2.15	2.02	1.98	2.05
Ridge	4.75	4.58	4.57	4.63
GP	4.37	4.20	4.15	4.24
KNN	7.60	8.73	8.61	8.33
Torus	4.73	4.69	4.20	4.55
ANN	4.45	4.13	3.97	4.19

Table 4.3: Predicted performance on test sets when temperature forecasts with $\sigma_\epsilon^2 = 0.063$ are used: yearly RMSE (MSCM) of the five forecasters.

In the second session, the predictions of table 4.3 were validated by comparing them with the RGD forecasting RMSE achieved using temperature forecasts. As it can be seen in table 4.4, the actual RMSE are in good agreement with their predictions. This can also be visually appreciated in fig. 4.8, where theoretical predictions are plotted against the actual RMSE. Again, ANN and GP are the best performers, closely followed by the Ridge and Torus forecasters, while KNN is the worst RGD predictor.

Year	2015	2016	2017	2015-2017
Ridge	4.68	4.28	4.28	4.42
GP	4.25	4.12	4.07	4.15
KNN	7.35	8.55	8.37	8.11
Torus	5.40	4.33	3.96	4.60
ANN	4.34	4.10	3.64	4.04

Table 4.4: Performance on test sets: yearly RMSE (MSCM) of the five forecasters trained and tested using one-day-ahead forecasted temperatures.

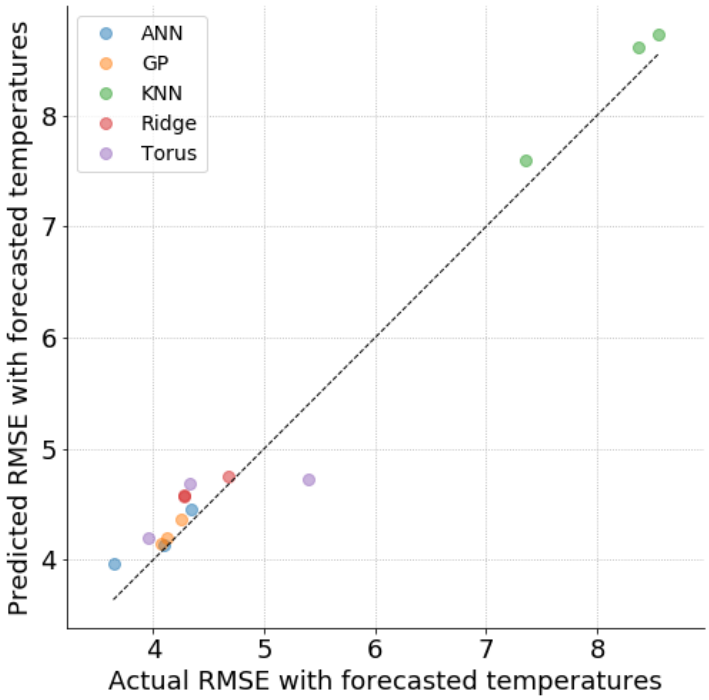


Fig. 4.8: Validation of the model predicting effects on gas forecast of temperature forecast errors. Gas forecast RMSE: theoretical prediction vs actual value.

A second assessment of the models was made in terms of their MAE. Hereafter, one-day-ahead forecasted temperatures are employed in the features. Results on the test sets are shown in table 4.5. Now, GP is the best performer, achieving an average MAE of 2.53 MSCM over the three test years. ANN, Torus, and Ridge Regression follow in the order. KNN is again the worst model, with an average MAE of 5.05 MSCM.

Year	2015	2016	2017	2015-2017
Ridge	3.39	3.10	3.01	3.17
GP	2.60	2.48	2.51	2.53
KNN	4.57	5.51	5.08	5.05
Torus	3.18	2.66	2.55	2.80
ANN	2.76	2.68	2.43	2.62

Table 4.5: Yearly MAE (MSCM) on test sets.

The differences between the RMSE- and MAE-based rankings are possibly explained by the non-Gaussianity of the prediction errors. In case of zero-mean prediction errors that are perfectly Gaussian, it should be MAE/RMSE $= \sqrt{2/\pi} \sim 0.798$, yielding identical rankings, irrespective of the adopted metrics. It occurs that MAE/RMSE < 0.798 for all the models: the ratio MAE/RMSE is about 0.61 for GP and Torus, 0.62 for KNN, 0.65 for ANN and 0.72 for Ridge. This is explained by the non-Gaussianity of the prediction errors, possibly associated with the presence of "fat tails" in their distributions. In particular, from fig. 4.9 it is apparent that different error variances are observed in the cold and warm seasons. This means that the overall error distribution is akin to a mixture, which can produce fat tails when the variances in the two seasons are much different. The error distributions in 2017 are displayed in fig. 4.10.
Due to the seasonal behavior of RGD, it is of interest to disaggregate data at a monthly level. In table 4.6, the monthly averages of MAE and MAPE are reported throughout the 2015-2017 test years. It appears that GP is the best performer during the warm period, especially from June to October, whereas in the cold months, from December to February, ANN is more accurate. A possible explanation is that the GP model is better at capturing the effects of the weekly seasonality, that explains most of the Summer variability, while ANN better allows for the non-linear effect of temperature, mostly relevant during the cold months.
To the best of our knowledge, there are no published benchmarks for the forecasting task addressed in this section. A somehow similar problem was studied by Zhu et al. [41] relative to UK gas demand in 2012. Still, their results are not entirely comparable to those of this thesis, for two main reasons: first, the authors considered the total UK demand and not just the residential one; second, UK climate is colder than the Italian one. Nonetheless, we can use relative error metrics, such as the MAPE in order to obtain a first level comparison, limited to 6 cold months (from October to March). My best model in terms of average MAPE over 2015-2017, i.e., the GP, achieves 3.11%, while Zhu's false neighbors filtered-support vector regression local predictor (FNF-SVRLP) achieves 3.88% on the same six cold months of 2012. Although no definite conclusion can be drawn, these numbers suggest some degree of consistency between forecasting performances at country level.

	MAPE(%)					MAE (MSCM)				
Month	Ridge	GP	KNN	Torus	ANN	Ridge	GP	KNN	Torus	ANN
January	3.10	3.01	6.45	3.21	2.93	5.79	5.67	12.44	5.96	**5.45**
February	2.75	2.84	4.77	3.33	2.62	4.52	4.59	7.60	5.48	**4.33**
March	3.94	4.20	7.13	3.83	4.27	4.59	4.89	8.01	**4.56**	4.90
April	6.17	4.80	14.79	4.89	5.09	3.56	**2.89**	8.23	2.98	3.04
May	5.76	2.67	6.08	3.21	2.50	2.37	1.22	2.67	1.51	**1.14**
June	4.57	1.32	6.02	3.37	1.92	1.46	**0.43**	1.92	1.11	0.62
July	3.78	1.16	3.65	1.50	1.54	1.11	**0.35**	1.13	0.45	0.46
August	9.39	3.00	19.44	3.86	4.50	2.24	**0.71**	4.56	0.92	1.06
September	5.36	1.06	3.18	1.33	1.81	1.92	**0.38**	1.17	0.49	0.68
October	4.10	2.81	6.22	3.30	3.42	2.23	**1.78**	3.91	1.99	2.09
November	2.70	3.14	5.50	3.03	2.9	**3.39**	3.76	6.47	3.57	3.41
December	2.78	2.68	4.27	2.70	2.52	4.83	4.58	7.14	4.62	**4.32**

Table 4.6: Monthly MAPE and MAE (MSCM) on test sets 2015-2017: best performers in terms of MAE are highlighted in boldface.

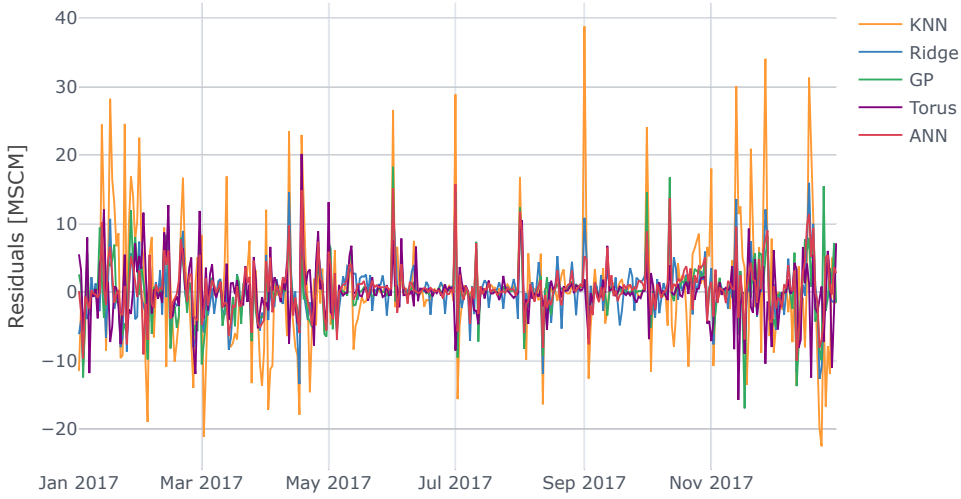


Fig. 4.9: Out-of-sample model residuals in 2017

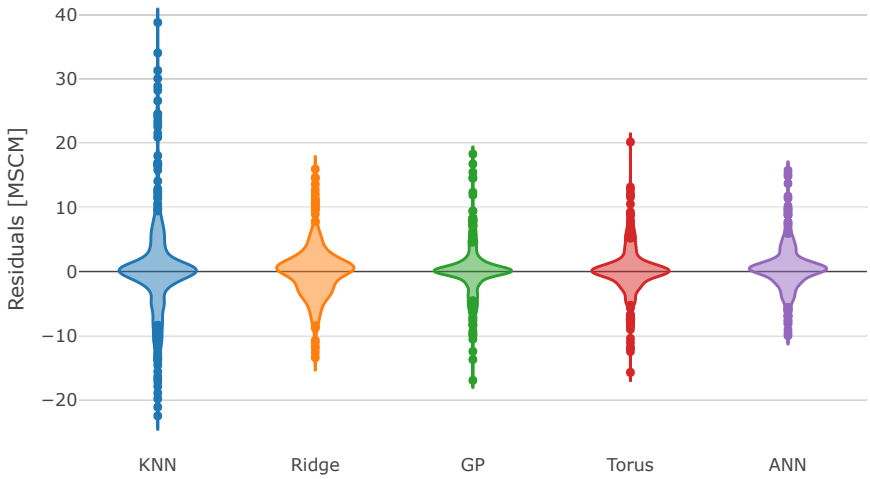


Fig. 4.10: Distribution of out-of-sample residuals in 2017

4.1.7 Conclusions

In this work, one-day-ahead forecasting of the residential gas demand was addressed at the country level. Five different models were developed and compared: Ridge regression, Gaussian Process, K-nearest neighbor, Artificial Neural Network, and the Torus model. The choice of the relevant covariates and the most relevant aspects of the preprocessing and feature extraction steps have been discussed, lending particular attention to the role of one-day-ahead temperature forecasts. In particular, a simple model describing the propagation of temperature errors to gas forecasting errors was derived.
The proposed methodology was tested on daily Italian gas demand data from 2007 to 2017. Although a specific benchmark is not available, a comparison with UK data restricted to cold months shows a substantial consistency between the performances achieved in the two countries.
Our best model, in terms of RMSE, was the Artificial Neural network, closely followed by the Gaussian Process. If the MAE is taken as an error measure, the GP became the best model, although by a narrow margin. From the analysis of monthly performance, GP was found to be more accurate in tracking the weekly periodicity, which is predominant in the summer period, while the ANN accounted better for the non-linear influence of temperature, whose contribution is more significant during the winter period.
An interesting question is how much of the forecasting mean square error is ascribable to temperature forecasting errors. On the Italian data, we found that the MSE for the ANN

model passed from $\text{MSE}^2 = 3.65^2 = 13.32$ (using true temperatures, see table 4.2) to $4.04^2 = 16.32$ (using temperature forecasts, see table 4.4). This means that temperature forecast errors account for some 18% of the RMSE of RGD forecasts. As demonstrated in fig. 4.8, our error propagation model successfully predicted the quantitative impact of temperature forecast errors on gas forecast errors, a capability that could prove useful in order to assess the extent and convenience of improvement margins associated with more sophisticated (and possibly more expensive) weather forecasts.

4.2 Short-Term forecasting of Italian gas demand

4.2.1 Introduction and literature review

Natural gas is one of the most important energy sources in Italy: it feeds domestic and industrial heating, production processes and thermoelectric power plants. Data from SNAM Rete Gas, the Italian Transmission System Operator (TSO), show that the total Gas Demand (GD) is made of three main components: Residential Gas Demand (RGD), Industrial Gas Demand (IGD), and Thermoelectric Gas Demand (TGD). In 2018, RGD accounted for 41.5% of the total consumption, IGD for 25.4% and TGD for the remaining 33.1% [52].

Accurate forecasts of the overall GD, as well as of its three main components, are of primary importance to energy providers, in order to improve pipe reservation and stock planning and also prevent financial penalties due to network unbalance. Moreover, GD is closely correlated with natural gas price, which is a key input for determining the optimal production plan of thermal power plants.

Several works addressed the forecasting of natural gas demand: comprehensive reviews are [3] and [2]. The latter proposes a classification along four dimensions: geographical area, time horizon, method, and inputs. Herein, we are interested in country-level, one-day-ahead predictions, based on statistical learning models that leverage past gas demand, temperature and calendar features as input variables.

With respect to prediction horizon, gas demand forecasting is usually divided into long-term forecasting, featuring an horizon of months or years and short-term, with an horizon of one or few days. Focusing on country-wide predictions, a long-term model based on temperature was proposed in [53] to forecast Turkish demand. The importance of the relation between weather and gas demand is also highlighted in [49] and [39]. In [38] a statistical model was applied to forecast the long-term evolution of Slovenian demand, while different kinds of so-called "grey models" were applied in [54] and [55] to forecast Chinese demand.

In [41], short-term forecasting of UK natural gas demand was addressed using support vector regression with false neighbours filtered. According to the authors, the method performed better than Auto-Regressive Moving Average (ARMA) models and neural networks (ANN). Azadeh et al. [40] proposed an adaptive network-based fuzzy inference system (ANFIS) to predict Iranian gas demand, which improved on classical time series methods and ANN. A more advanced model, combining wavelet transform, genetic algorithm, ANFIS and ANN was applied in [42] to the Greek gas distribution network.

Long-term evolution of the Italian gas demand is investigated in [56] and [57]: macroeconomic indicators, such as gross domestic product and gas prices and climatic factors

are used to build scenarios of RGD and overall GD evolution up to 2030. However, to the best of my knowledge, the daily series of Italian GD and its peculiar features have not been studied in the literature and no result about its short-term forecasting has been presented.
In previous section, focusing on Italian RGD, we proposed and compared five prediction models: ridge regression, Gaussian Process (GP), nearest neighbours, Artificial Neural Networks (ANN), and torus model, concluding that ANN and GP provided the best results. Herein, the analysis along three directions is extended: first, also the prediction of IGD and TGD is addressed, thus enabling the prediction of the overall Italian GD; second, four additional base forecasters (LASSO, elastic net, random forest, and support vector regression) are considered; third and finally, the use of ensemble predictors, i.e. forecasters obtained by the suitable aggregation of base forecasts, is investigated. More precisely, based on the nine base models, four ensemble predictors are considered (section 3.4): simple average, weighted average, subset average, and support vector regression aggregation.
Ensembling, also known as blending, is known to be an effective technique to improve overall accuracy and stability, see e.g. [58, 59]. Recently, ensemble predictors have been proven successful in forecasting electric load [60], whose series shows a periodic structure similar to the one of GD.
The contribution of this section and the related paper to literature is thus threefold: first, the statistical properties of Italian IGD and TGD are presented and discussed; on these data, we develop, apply, and compare nine machine-learning models; finally, the use of ensemble predictors is explored, assessing the consequent improvements.
This chapter is organised as follows. In section 4.2.2 the forecasting problem and the available data are presented, while in section 4.2.3 we describe the most relevant features of IGD and TGD time series. After discussing feature engineering (section 4.2.4), in section 4.2.5 the adopted models are introduced and training and hyperparameter tuning are presented in the section. Results are reported in section 4.2.6 and some concluding remarks (section 4.2.7) end the section.

4.2.2 Problem statement

In this chapter, the prediction of the Italian daily GD is addressed both at aggregated and disaggregated level: for each day, the overall GD is given by the sum of RGD, IGD, and TGD.
The datasets for RGD, IGD, and TGD are 12 year long, ranging from 2007 to 2018, and consisting of 3 fields: date (t), forecasted average temperature in Northern Italy (T)[1], and gas demand (RGD, IGD and TGD). Temperature in Northern Italy was considered as this region has the most rigid climate, and is thus more sensible to heating requirements. In fig. 4.11 the complete series of RGD, IGD, TGD and overall GD are displayed.

[1] Weather forecast were provided by one of the most known and specialised Italian company for these data.

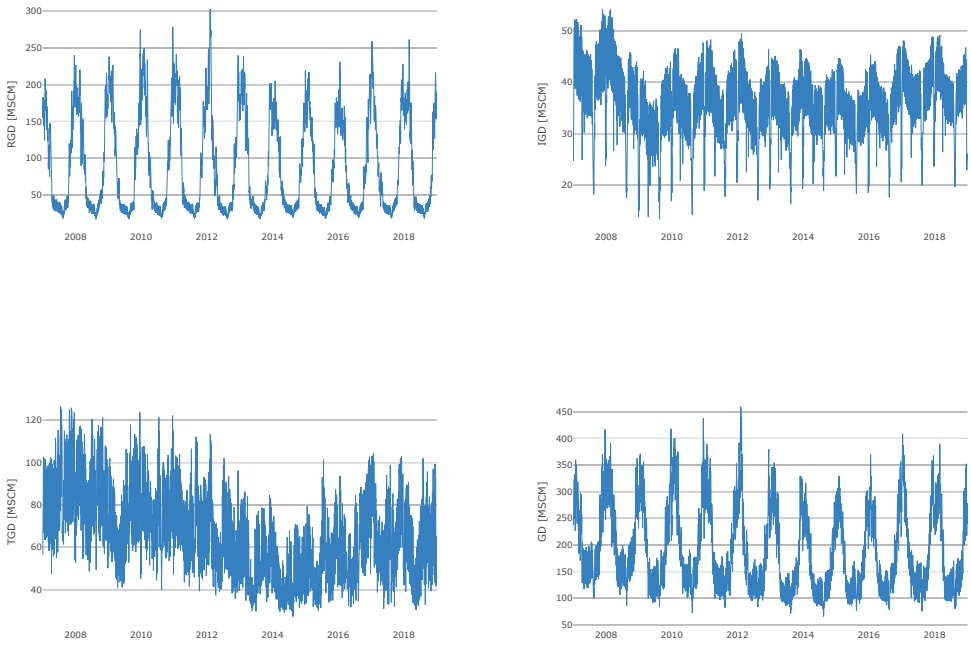


Fig. 4.11: Top left: Italian Residential Gas Demand (RGD); top right: Italian Industrial Gas Demand (IGD); bottom left: Italian Thermoelectic Gas Demand (TGD); bottom right: overall Italian Gas Demand (GD = RGD + IGD + TGD).

4.2.3 Exploratory analysis

For what concerns the general considerations regarding the gas demand time series we refer to section 2.3, where these series are widely described. Instead herein the analysis done of RGD, shown in section 4.1.3, is extended to IGD and TGD.

4.2.3.1 Industrial gas demand

The periodogram, plotted in fig. 4.12, exhibits peaks at periods of 365.25 and 7 days, while other relevant values are ascribable to multiple harmonics of the fundamental ones. Notably, differently from what happens for RGD (fig. 4.3), the weekly seasonality prevails on the yearly one in terms of magnitude.
Temperature is known to be a major determinant of gas demand [2, 39, 49]. In order to take into account that the need for heating ceases when temperature raises above 18 °C, it is useful to refer to the so-called Heating Degree Days (HDD), defined as $\mathrm{HDD} = \max(18 - T, 0)$, where T is the temperature in degrees Celsius. The scatter plots of IGD against temperature and IGD against HDD are reported in fig. 4.13.

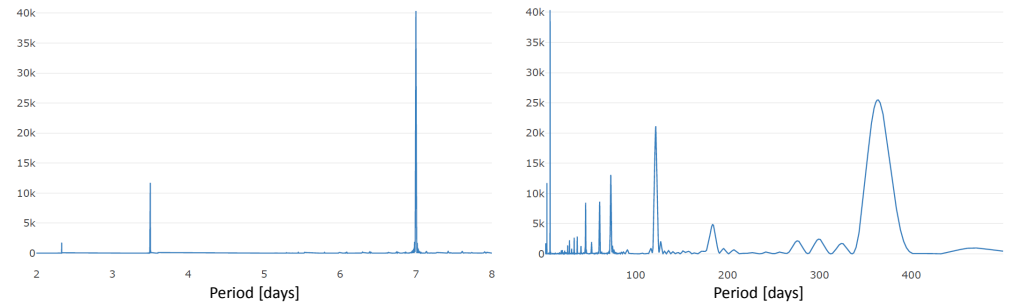


Fig. 4.12: IGD periodogram. Left panel: periods from 0 to 8 days; right panel: periods from 0 to 500 days.

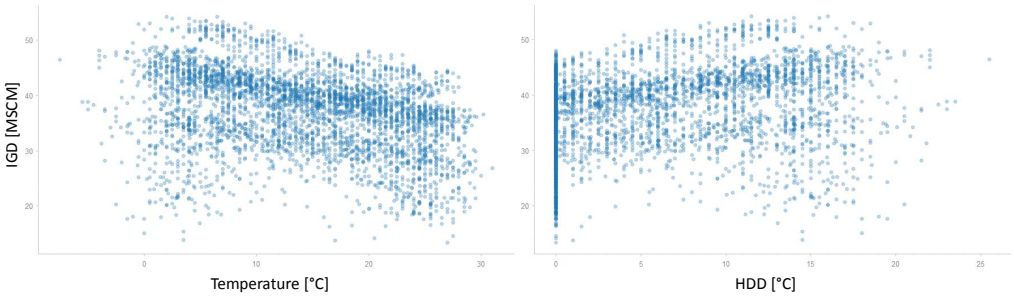


Fig. 4.13: Effect of temperature on IGD. Left panel: IGD vs temperature; right panel: IGD vs Heating Degree Days (HDD).

4.2.3.2 Thermoelectric gas demand

The periodogram in fig. 4.14 shows that, also for TGD, the main seasonal component is the weekly one, which is consistent with Italian power demand [61].
The scatter plot of TGD against temperature, displayed in the left panel of fig. 4.15, shows a peculiar U-shaped pattern: TGD increases as weather gets colder, but also when it gets hotter. In fact, in summer more thermoelectric production is required because air conditioning pushes the demand for electric power. This U-shaped pattern justifies the introduction of a suitable feature variable, herein named Heating Cooling Degree Day (HCDD). More precisely

$$\mathrm{HCDD} = |T_c - T|$$

We found that $T_c = 16°\mathrm{C}$ maximises the linear correlation between TGD and HCDD.

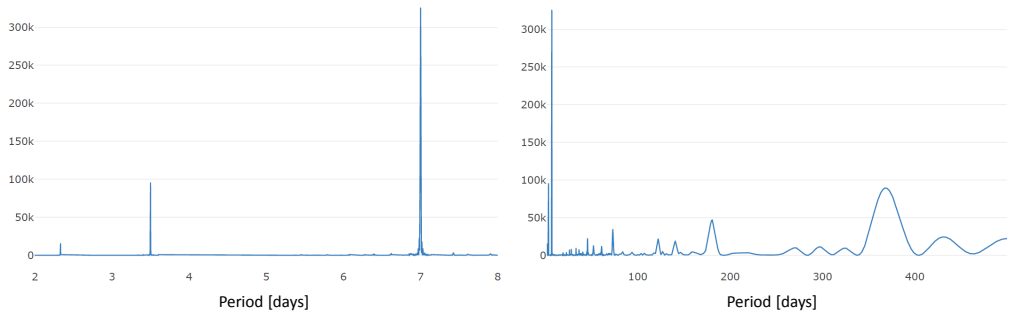


Fig. 4.14: TGD periodogram. Left panel: periods from 0 to 8 days; right panel: periods from 0 to 500 days

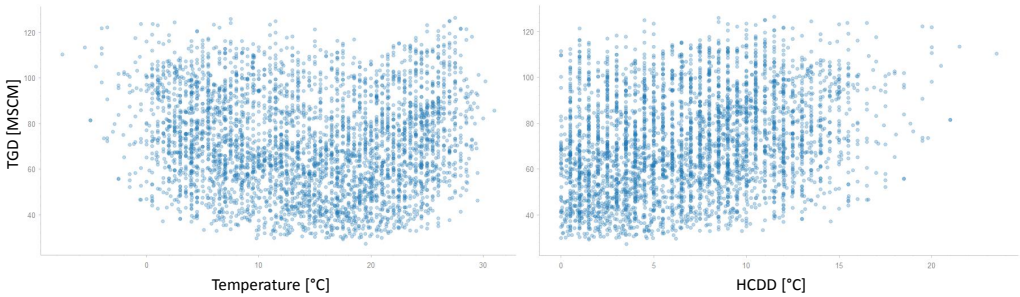


Fig. 4.15: Effect of temperature on TGD. Left panel: TGD vs temperature; right panel: TGD vs HCDD.

4.2.4 Feature extraction

Based also on the exploratory analysis, the features used in the prediction algorithms include: autoregressive terms, calendar features, temperature and its derived variables HDD and HCDD. Table 4.7 reports the complete list of features.
To predict $y_t, y \in \{\text{RGD}, \text{IGD}, \text{TGD}\}$, we included, as autoregressive features, y_{t-1}, y_{t-7}, $y_{\text{sim}(t)}$ and $y_{\text{sim}(t-1)}$.
As calendar features, binary dummy variables were introduced to account for weekdays and holidays. Dummy variables were also added to identify: (i) extended holidays, i.e. working days preceded and followed by either Saturdays, Sundays or holidays, and (ii) days after holidays, i.e. working days which immediately follow a holiday and are not extended holidays.
As temperature features we selected also forecasted temperatures T_t, T_{t-1}, T_{t-7} and $T_{\text{sim}(t)}$. In view of what shown in sections 4.1.3, 4.2.3, for both RGD and IGD, also HDD values at the same times were introduced, while, for TGD, HCDD replaced HDD.

4.2.5 Predictive models and implementation notes

We tested all the nine base models described in section 3.2, which can be grouped into three categories:

1. linear models: ridge regression, lasso, Torus model [61], support vector regression, and elastic net
2. non-linear models: random forest, neural networks
3. non-parametric models: Gaussian Process, nearest neighbour

Feature	Reference time	Type	Series
Gas demand series	t-1	continuous	RGD, IGD, TGD
Gas demand series	t-7	continuous	RGD, IGD, TGD
Gas demand series	sim(t)	continuous	RGD, IGD, TGD
Gas demand series	sim($t-1$)	continuous	RGD, IGD, TGD
Forecasted temperature	t	continuous	RGD, IGD, TGD
Forecasted temperature	t-1	continuous	RGD, IGD, TGD
Forecasted temperature	t-7	continuous	RGD, IGD, TGD
Forecasted temperature	sim(t)	continuous	RGD, IGD, TGD
Forecasted HDD	t	continuous	RGD, IGD
Forecasted HDD	t-1	continuous	RGD, IGD
Forecasted HDD	t-7	continuous	RGD, IGD
Forecasted HDD	sim(t)	continuous	RGD, IGD
Forecasted HCDD	t	continuous	TGD
Forecasted HCDD	t-1	continuous	TGD
Forecasted HCDD	t-7	continuous	TGD
Forecasted HCDD	sim(t)	continuous	TGD
Weekday	t	categorical	RGD, IGD, TGD
Holiday	t	dummy	RGD, IGD, TGD
Day after holiday	t	dummy	RGD, IGD, TGD
Bridge holiday	t	dummy	RGD, IGD, TGD

Table 4.7: List of features

Four of them, namely ridge regression, Gaussian Process (GP), Torus model, nearest neighbours and neural networks, were already applied to RGD in section 4.1.
Moreover, we tested four ensemble models, described in section 3.4, which aggregate forecasts issued by the basic models: (i) Simple average, (ii) Weighted average, (iii) Subset average (b.f.), and (iv) SVR aggregation.
The available data range from 2007 to 2018. Four one-year long test sets, ranging from 2015 to 2018, were used to obtain a comparative assessment of the 13 models, including 9 base models and 4 ensemble ones. Each test set was associated to a set of training data, that were organised differently depending on the nature of the considered model, either base or ensemble.
Training of base models. The training set, called *base training set* $\mathcal{T}_{\text{base}}(\mathcal{Y})$, is made of all data previous to the test year $\mathcal{Y}$. For instance, if $\mathcal{Y} = \{2017\}$ is taken as test set, the 9 base models were trained on the *base training set* $\mathcal{T}_{\text{base}}(\{2017\}) = \{2007, \dots, 2016\}$.
Training of ensemble models. In this case two training sets were considered. The year before the test set $\mathcal{Y}$ was used as *ensemble training set* $\mathcal{T}_{\text{ens}}(\mathcal{Y})$, while the remaining data were used to train the 9 base models that enter the aggregation. For instance, if $\mathcal{Y} = \{2017\}$ is taken as test set, the 9 base models were trained on $\mathcal{T}_{\text{base}}(\{2016\}) = \{2007, \dots, 2015\}$, while the ensemble models were trained on $\mathcal{T}_{\text{ens}}(\{2017\}) = \{2016\}$.
Hyperparameters of the Torus model were tuned by maximising AIC, those of the Gaussian Process by maximising the marginal likelihood, while for all the other base models five-fold cross validation was used.

Test of base models. For the test set $\mathcal{Y}$, base predictions were computed using the base models trained on $\mathcal{T}_{\text{base}}(\mathcal{Y})$.
Test of ensemble models. Given the forecasts provided by the base models trained on $\mathcal{T}_{\text{base}}(\mathcal{Y})$, the ensemble forecasts were obtained using ensemble models trained on $\mathcal{T}_{\text{ens}}(\mathcal{Y})$.
Out-of-sample performances were evaluated in terms of Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t|$$

where y_t and $\hat{y}_t$, $y \in \{\text{RGD}, \text{IGD}, \text{TGD}, \text{GD}\}$, are the actual value and its forecast, while n is the number of samples in the considered test set. MAE was preferred over percent or relative error metrics due to the large range of values assumed by the target variable, which would give undue importance to poor performances during low-demand periods.

4.2.6 Results

Mean absolute errors for RGD, IGD, TGD, and total GD are reported, respectively, in Tables 4.8, 4.9, 4.10, and 4.11.
For what concerns base models, GP, ANN and SVR achieved the best average MAE across all the gas demands, with differences between each other smaller than 0.10 MSCM. Notably, results achieved by such models were also stable across different test sets. On the other hand, KNN was consistently the worst performer, due to its poor capability of modelling influence of temperature and holidays.
Ensemble models consistently outperformed base ones: in particular, subset average achieved the best average MAE on all four types of gas demand: the three disaggregated demands and the total one. A possible explanation is that different models are better at capturing specific behaviours: in section 4.1, for instance, it was shown that the ANN model achieved the best results in winter, while GP in summer, suggesting that the former is better at modelling the impact of weather, while the latter can better follow seasonal patterns. Aggregation can indeed mitigate errors committed by single models, thus increasing overall accuracy and robustness.
The improvement due to aggregation was particularly evident for RGD (table 4.8), where the best base model (GP) was outperformed by the best ensemble model (subset average) by 0.36 MSCM. The gap between base and ensemble models was smaller for the other two gas demands: GP and SVR are worse than subset average by 0.07 MSCM, for IGD (table 4.9); SVR is worse than subset average by 0.09 MSCM, for TGD (table 4.10). Finally SVR is worse than subset average by 0.29 MSCM, for the global Italian GD (table 4.11).
The 2018 forecasts and the corresponding residuals provided by the best ensemble predictor, namely subset average, are displayed in fig. 4.16 and fig. 4.17, respectively.
To the best of our knowledge, the only term of comparison available for the task addressed in this section is given by the forecasts of the global Italian GD issued by SNAM Rete Gas, the Italian Transmission System Operator (TSO) [62]. In 2017 and 2018, the improvement is neat: the out-of-sample MAE of SNAM predictions was 9.62 MSCM in 2017 and 8.30 MSCM in 2018, while our best model (subset average) scored 5.16 MSCM in 2017 and 5.46 MSCM in 2018, see table 4.11.

Model	2015	2016	2017	2018	Average
Ridge	3.30	3.06	2.95	3.55	3.21
Lasso	3.30	3.06	2.95	3.56	3.22
Elastic net	3.30	3.06	2.95	3.56	3.22
SVR	2.84	2.62	2.38	2.93	2.69
GP	2.66	2.59	2.57	2.83	2.66
KNN	4.68	5.47	5.05	5.65	5.21
Random forest	3.04	3.36	3.50	3.48	3.35
Torus	3.18	2.66	2.54	3.28	2.91
ANN	2.76	2.68	2.43	3.10	2.74
Simple average	2.66	2.57	2.45	2.91	2.65
Subset average (b.f.)	**2.41**	**2.17**	**2.06**	**2.56**	**2.30**
Weighted average	2.59	2.33	**2.06**	2.64	2.40
SVR aggregation	2.58	2.30	2.19	2.67	2.44

Table 4.8: Forecasted Residential Gas Demand: out-of-sample MAEs. Each year's best performers are in boldface.

Model	2015	2016	2017	2018	Average
Ridge	0.75	0.75	0.74	0.77	0.75
Lasso	0.75	0.75	0.74	0.77	0.75
Elastic Net	0.75	0.75	0.74	0.77	0.75
SVR	0.57	0.58	0.7	0.75	0.65
GP	0.61	0.61	0.68	0.70	0.65
KNN	1.46	1.25	1.95	1.23	1.47
Random Forest	0.78	0.86	0.95	0.83	0.86
Torus	0.96	0.97	1.05	1.10	1.02
ANN	0.66	0.80	**0.57**	0.74	0.69
Simple average	0.60	0.62	0.69	0.66	0.64
Subset average (b.f.)	0.56	0.56	0.58	**0.61**	**0.58**
Weighted average	**0.55**	**0.55**	0.65	0.70	0.61
SVR aggregation	0.57	0.79	**0.57**	0.81	0.68

Table 4.9: Forecasted Industrial Gas Demand: out-of-sample MAEs. Each year's best performers are in boldface.

Model	2015	2016	2017	2018	Average
Ridge	3.73	4.15	4.26	4.48	4.15
Lasso	3.73	4.15	4.26	4.49	4.16
Elastic Net	3.73	4.15	4.26	4.49	4.16
SVR	3.41	3.64	4.33	4.33	3.93
GP	3.49	3.70	4.39	4.34	3.98
KNN	6.13	5.22	5.83	5.54	5.68
Random Forest	4.66	4.43	4.87	4.84	4.70
Torus	3.98	4.48	4.96	4.94	4.59
ANN	3.40	3.97	4.32	4.41	4.03
Simple average	3.50	3.75	4.21	4.36	3.96
Subset average (b.f.)	**3.26**	3.65	**4.17**	**4.26**	**3.84**
Weighted average	3.35	3.71	4.31	4.31	3.92
SVR aggregation	3.38	**3.62**	4.28	4.37	3.91

Table 4.10: Forecasted Thermoelectric Gas Demand: out-of-sample MAEs. Each year's best performers are in boldface.

Model	2015	2016	2017	2018	Average
Ridge	6.32	6.34	5.80	6.57	6.26
Lasso	6.32	6.34	5.81	6.57	6.26
Elastic Net	6.32	6.35	5.81	6.57	6.26
SVR	5.23	5.05	5.55	5.85	5.42
GP	5.33	5.23	5.88	5.82	5.57
KNN	9.04	9.31	9.97	9.83	9.54
Random Forest	6.58	6.45	7.15	7.11	6.82
Torus	6.56	6.47	6.40	7.00	6.61
ANN	5.43	5.50	5.47	6.08	5.62
Simple average	5.53	5.40	5.56	5.98	5.61
Subset average (b.f.)	**5.02**	**4.80**	**5.23**	**5.46**	**5.13**
Weighted average	5.27	5.01	5.34	5.55	5.29
SVR aggregation	5.19	4.91	5.29	5.79	5.30
SNAM forecast	n.a.	n.a.	9.62	8.30	n.a.

Table 4.11: Forecasted Italian Gas Demand: out-of-sample MAEs. Each year's best performers are in boldface.

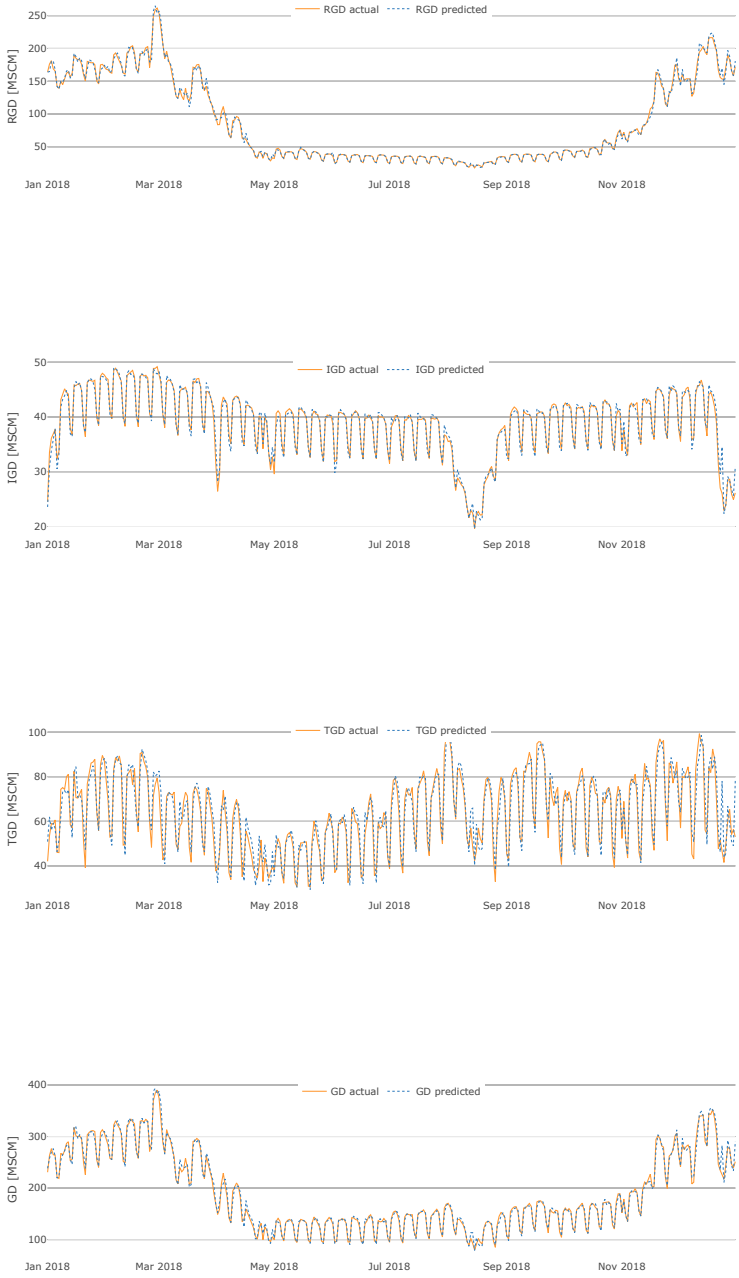


Fig. 4.16: Subset average (b.f.): predicted gas demands in 2018. From top to bottom: Residential Gas Demand (RGD), Industrial Gas Demand (IGD), Thermoelectic Gas Demand (TGD), overall Gas Demand (GD).

Fig. 4.17: Subset average (b.f.): one-day-ahead prediction residuals in 2018. From top to bottom: Residential Gas Demand (RGD), Industrial Gas Demand (IGD), Thermoelectic Gas Demand (TGD), overall Gas Demand (GD).

4.2.7 Conclusions

The industrial and thermoelectric components of Italian daily gas demand were analyzed, completing a previous study concerning residential demand. Industrial and thermoelectric demand were found to show different relationships with temperature and crafted features to properly take them into account.
Several forecasting models were investigated and compared: nine base models plus four ensemble models. Aggregated models were found to be consistently more effective than base ones. In particular, in 2017 and 2018 the best ensemble model, i.e. subset average, outperformed forecasts provided by the Italian TSO.

5

Italian power demand forecasting

5.1 Introduction and literature review

Electricity demand is a relevant quantity for the utilities in the energy sector, above all for the electricity producer companies that need to determine their electricity production one day ahead in order to better meet their client demand. On its side, electricity demand represents a fundamental regressor in the electricity price formation as well as a driving element in the power plant bidding strategy. As a consequence, both long-term and short-term load forecasting (LTLF and STLF) are necessary activities, and their results have a direct impact on the companies' performance.
In this chapter we focus on day-ahead Italian electricity demand (IED) forecasting, which is part of STLF.
There is a large set of literature on STLF, starting at least since 1918, as reported by three works [4],[5],[6] that compile a bibliography before the 1980. Works published between the 1980 and 2010 are well summarized in [7] and a more comprehensive and recent literature review is given by [8]. The principal elements where the papers differ are the modelling techniques adopted, the regressors considered and the electricity demand time series used for the STLF.
Regarding the models, historically time-series and state-space methods have been used [63] and preferred to the similar-day approach, where STLF is directly assigned as the same load of the most similar day in the past [64]. Multiple linear regression has also been extensively adopted [63],[65],[66],[67],[68] taking into account of the weather variables and their relation, linear and also non-linear, with the electricity demand, calendar variables and holiday modelling. Some papers have compared many methods such as stochastic time series, multiple linear regression, exponential smoothing and state space [69], double seasonal ARMA, an extension of Holt-Winters exponential smoothing for double seasonality, artificial neural networks (ANN), and PCA-based regression [70], ARIMA, periodic AR, double seasonal Holt-Winters exponential smoothing, an alternative exponential smoothing formulation and a PCA-based method [71]. The authors of [70] and [71] have highlighted how the double seasonal Holt-Winters exponential smoothing outperforms the other models on many different time series such as the electricity demand in Rio de Janeiro, England and Wales [70] and the electricity demand data from ten European countries [71].

Much literature is also devoted to the application of ANN to STLF, a comprehensive review being given by [72], where the authors have analysed the reasons behind the scepticism around these methods, mostly in virtue of their overparameterization with the risk of overfitting and the lack of appropriate analysis of test errors. The authors of [65] have also implemented an ANN technique comparing the results with multiple linear regression, showing that better performances are achieved by non-linear methods [73]. Further advances of ANN techniques for STLF have been carried out by a group of authors since 1995 to 2002 [74],[75],[76],[77],[78]. More recently, a multi-stage ANN has been introduced in order to improve the performance when weather forecast errors are larger [79].
Other approaches have been recently applied to STLF, such as support vector machines (SVM) [80],[81] and kernel based methods [82].
In view of the correlation between electric load and the weather conditions, among the potential regressors there are weather variables such as dry bulb temperature (the standard measure of temperature), wet bulb temperature, level of humidity, wind speed, dew point temperature, etc. The most commonly used, obviously, is the temperature at different past times as well as its forecast. Furthermore temperature can enter in the model as a naked variable or through a transformation - piecewise linear, quadratic, cubic, etc. - describing its non-linear relationship with the electricity demand [65],[83],[81].
As a consequence of the characteristic seasonal patterns, the calendar variables represent another set of typical regressors: month of year, season, weekday, holiday and day close to holidays.
Although the practical applications and the scientific literature on STLF cover a number of countries all over the world, the different peculiarities of each country, such as the geographical position and the use of electricity-powered air conditioners, imply that a single model cannot be suitable for each situation. Herein we focus on the Italian consumption, comparing our results with other recent works on this country [84],[85].
In this chapter, we describe how IED forecasts were obtained through the application of the machine learning models introduced in chapter 3 and already used for the forecast of the Italian gas demand (GD) in chapter 4. In particular, we consider nine different learning models - ridge, LASSO, elastic net, Torus, SVM, GP, KNN, random forest and ANN - and four ensemble techniques that aggregate the base forecasts - simple average, weighted average, a simple average based on optimized regressors and SVM. An aggregation model was also added, Subset average (c.a.), described in chapter 3.
The same regressors considered for the GD forecasting were used, while a different function describing the relation between IED and temperature is tested. Furthermore, a SARIMA model on the errors of each base prediction was applied in order to remove their residual autocorrelation and improve the forecast results. This correction was not necessary in any case of gas demand forecasting, presented in chapter 4, because those forecasts did not show residual autocorrelation.
This framework was applied to the daily electricity demand time series using the daily temperature forecast as covariate. The hourly forecast of IED was approached in much the same way: in a first step, following the hourly multiple model approach of [83], the model procedure applied to the daily IED is repeated 24 times, one for each hours, with the use of 24-hour ahead temperature forecasts. Then, the modeling was also repeated for each of the 6 zones and each of the 24 hours, thus iterating the procedure 144 times. Finally, dimensionality reduction via principal component analysis (PCA) was performed

in order to reach better performances in terms of computation time and possibly reduce overfitting. In each of these experiments we are interested in analyzing the benefit of aggregation finalized to obtain the prediction of daily or hourly IED.
This chapter is organized as follows: in section 5.2 the main objective is described, in section 5.3 all the analysis on IED and the chosen regressors are detailed, section 5.4 is devoted to the models and methods adopted. In section 5.5 our application of the models to IED is explored; in section 5.6 the final results are reported and discussed. Lastly, in section 5.7 some conclusion are drawn and future developments are discussed.

5.2 Problem Statement

The scope of this section is the extension of the machine learning model architecture, used in in chapter 4 for the forecast of the Italian gas demand (GD), and its application to day-ahead Italian electricity demand forecasting with the consequent analysis of its forecasting results.
IED time series, reported in fig. 2.14 during the period 2012-2018, shows characteristics similar to those highlighted by the Italian gas demand series Figs. 2.2, 2.3 and 2.4, such as pronounced periodicities and high correlation with temperature. This observation opens the opportunity to test the same models as those developed for GD.
The forecast errors resulting from the model application are analyzed and corrected by the application of a time-series model - SARIMA - so as to withen the residuals.
This forecasting procedure is followed for both the hourly and daily data in order to obtain the hourly and aggregated daily forecasts. Our first scope is to pass through the hourly forecasting and then obtain the daily forecasts by aggregating the hourly results, assessing the value added achieved by this solution. Second, PCA was applied to reduce the number of hourly components to forecast and, after that, we resort to the 24 hour IED and its aggregation to daily IED. These two passages, namely disaggregation and dimensionality reduction via PCA, are applied also to the time series of the different Italian zones. Finally, the results with and without PCA are compared.

5.3 Italian power demand analysis

In this section we follow the data analysis shown in chapter 3 in relation to the daily IED. We start by exploring all the characteristics of IED, the target time series. Its relation with the principal exogenous predictive regressor, the temperature, is studied and the list of chosen regressors is described. Then the analysis is extended by also considering the single 24 hour IEDs with daily frequency and their relation with the daily IED and hourly temperature. Finally we introduce the spatial disaggregation of IED in the 6 zonal Italian Electricity Demands, XED, where a generic X stands for each zone.

5.3.1 IED time series analysis

As highlighted in fig. 5.1, IED is strongly autocorrelated, above all with a 7-day time lag, with correlation around 0.78, larger than the one-day lag correlation which is about 0.6. This behaviour is also seen looking at fig. 2.14 where the weekly shape repeates

over time while the one-day differences strongly depend on the day of the week under question. As a matter of fact, in the period 2012-2018, the average percent differences between consecutive days are characterised by a strong growth of 28% between Sunday and Monday, low variations between the following days until Friday and two pronounced decreases, between Friday and Saturday and between Saturday and Sunday, respectively of 15% and 12%.

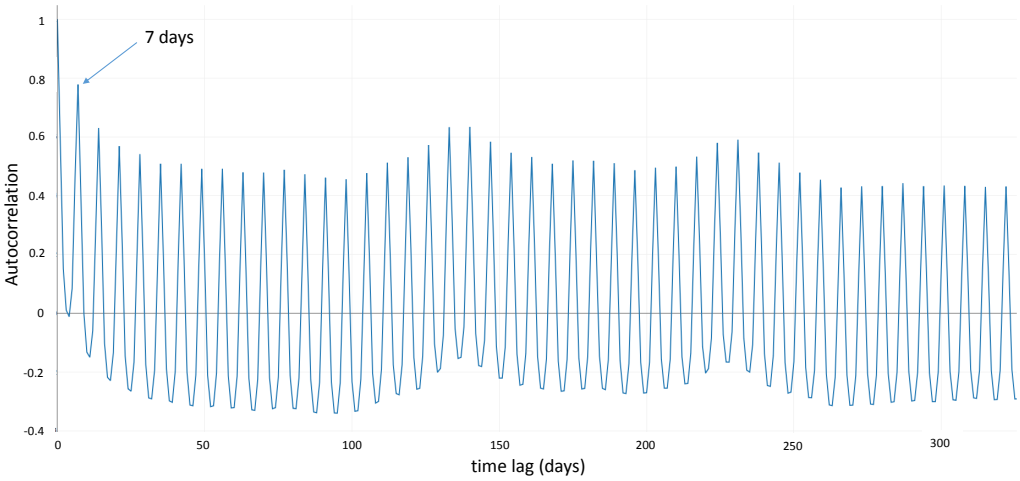


Fig. 5.1: IED autocorrelation function computed on 2012-2018 data. The seven day lag correlation is evident.

The analysis of the spectral density, reported in fig. 5.2, clearly shows that the weekly periodicity is the most evident. There is a yearly periodic signal, with the demand level rising in correspondence of the coldest and hottest periods, respectively, of winter and summer, generally in February and July. All the other spikes in the spectral density are ascribable to multiple harmonics of the fundamental ones.

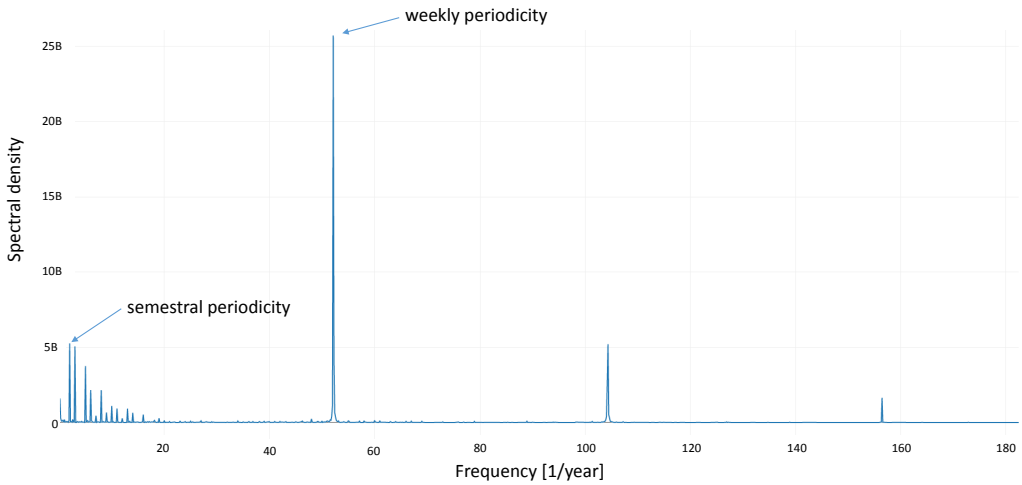


Fig. 5.2: IED spectral density function computed on 2012-2018 data. The spectral density is moreover concentrated on the point with frequency 52/year corresponding to the weekly periodicity.

As a consequence of the seasonalities, IED also shows a non-negligible correlation with its value one year before, about 0.55. This correlation rises to 0.88 if similar days are considered, as defined in [86] which can be summarized as the closest day, in terms of daynumber 1-365, that in the previous year shares the same weekday. In case of a holiday, the similar day is just the same holiday in the previous year.

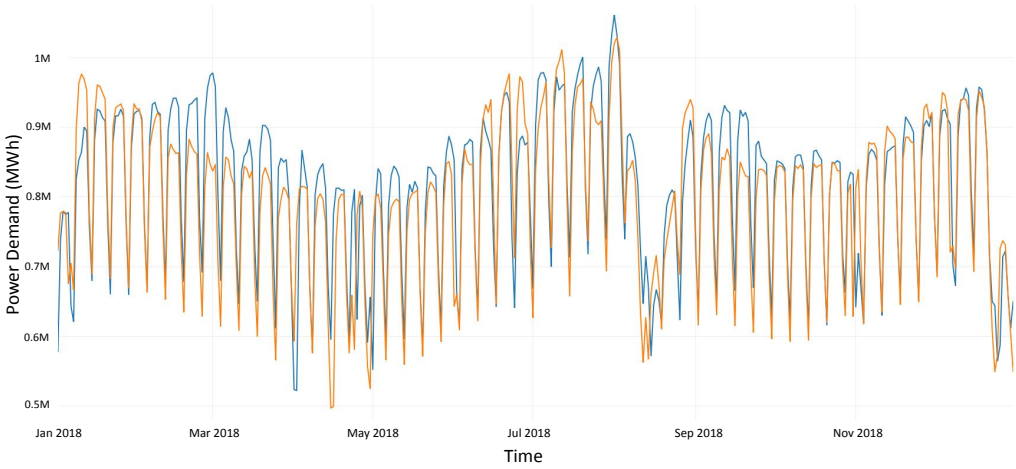


Fig. 5.3: IED of year 2018 (blue) versus similar day IED of year 2017 (orange)

5.3.2 IED vs Temperature

An exogenous regressor is given by the temperature, for which value we use the Italian aggregation supplied by one of the best-known providers of weather and weather-related data in the energy sector.
The relation between IED and temperature is well represented in fig. 5.4 where the U-shape function may be justified by cooling and heating needs respectively in case of high and low temperatures. This behaviour, well known in the literature [87],[65], partially depends on the country under consideration and its level of air conditioner usage as well as electric rather than gas heating. Figure 5.4 shows a critical temperature (T_c), point of inversion of correlation between IED and temperature, around 15°C. Furthermore the U-shape relation is notably asymmetric with a steeper slope for temperatures higher than T_c compared with the other branch of the curve, possibly indicating a greater usage of air conditioning than electric heating. Finally, in fig. 5.4 the different levels of electric demand between weekdays, Saturday and Sunday are shown with a similar level from Monday to Friday, a lower demand on Saturday and the lowest on Sunday. Under the Sunday level there are points with different colours that represent holidays.

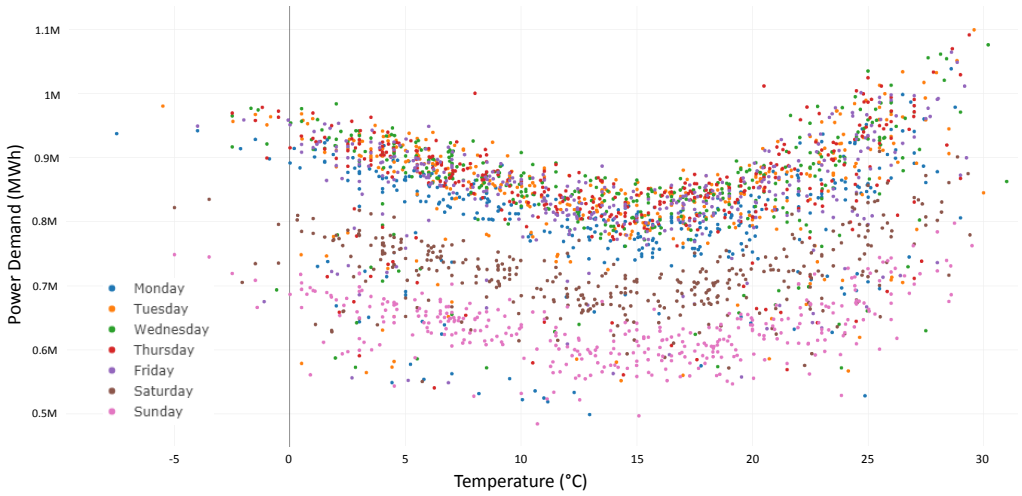


Fig. 5.4: IED versus temperature: an asymmetric U-shape behavior at different levels depending on the days of week.

The non-linear relation between IED and temperature can be captured via the HCDD (Heating and Cooling Day Degrees) given by

$$\mathrm{HCDD} = |T_c - T| \,. \tag{5.1}$$

This function can be used to transform the weather temperature regressors, as done for Thermoelectric Gas Demand forecasting in [11]. $\mathrm{T_c}$ is the critical temperature and corresponds to the point of the inversion of the slope.
The correlation between IED and HCDD is about 0.3 during the weekdays and grows to 0.4 for Saturdays and 0.5 for Sundays. The real correlation could also be higher but turns out to be lowered by the effect of holidays and business closure periods especially in August and during the Christmas vacation.

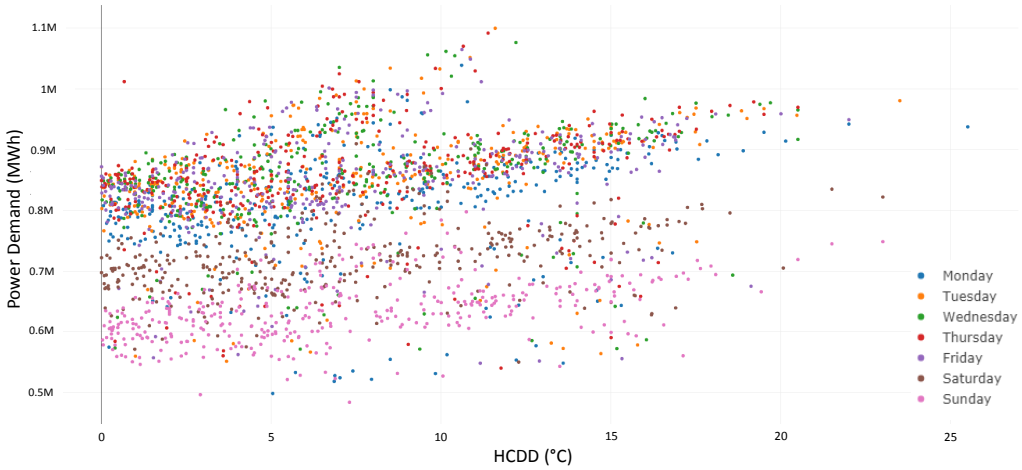


Fig. 5.5: IED versus HCDD: a linear relation at different levels depending on the day of week.

Another possible solution that takes account of the asymmetry of the U-shape relation between IED and temperature is to consider two distinct day degree regressors: Heating Degree Day (HDD)

$$\mathrm{HDD} = \max(T_c - T, 0) \tag{5.2}$$

and Cooling Degree Day (CDD)

$$\mathrm{CDD} = \max(T - T_c, 0) \tag{5.3}$$

where $\mathrm{T_c}$ is the same in both definitions. In this way the models are allowed more flexibility in order to learn the patterns driven by temperature.
The temperature transformation should take into account the day-ahead change of temperature, since the most important effect, given by the level of temperature, is already embedded in the power demand of the day before. This is shown in fig. 5.6, where the right-hand panel displays the relation between IED(t-1) and temperature(t), very similar to the one between IED(t) and temperature(t), while the left scatter plot of IED(t) against IED(t-1) highlights their strong linear relation.

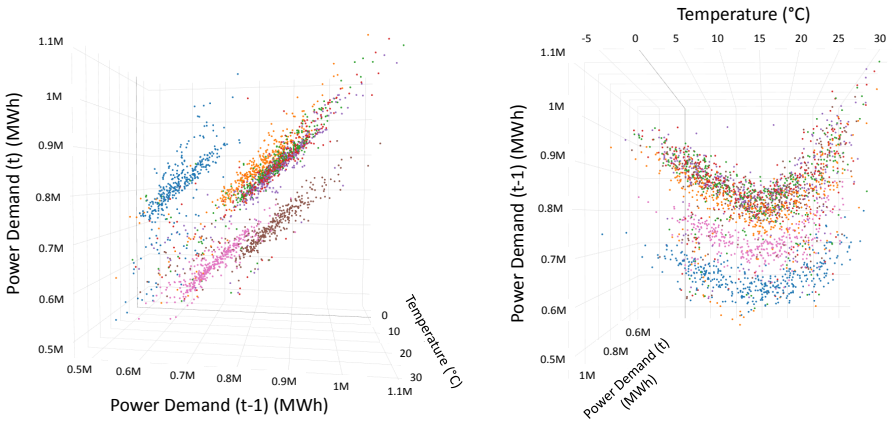


Fig. 5.6: A 3d representation of IED(t) respect IED(t-1) and Temperature(t).

5.3.3 Hourly IED

IED can be disaggregated at time and spatial-time levels, respectively IED_h and XED_h. In view of the high correlations between the different IED_h and with IED_d, described in chapter 2, the behavior of IED_d is roughly replicated by each IED_h, just changing the level. This pronounced similarity suggested the possibility to treat IED_h with same models and regressors as IED_d, just replacing the daily temperature forecast with the hourly one.
As highlighted in chapter 2, similar considerations apply to each XED_h as well to the North series, more regular and correlated to IED_h, also because it covers about 56% of the total Italian demand. As a consequence the same models and features chosen for IED forecasting could be applied to the XED_h forecasting.
The relationship between IED_h and temperature at the corresponding hour can be divided into two different periods of the day: from 6 a.m. to 11 p.m., when the U-shape is similar to the daily one, and in the remaining hours when the left branch of the curve is much lower, probably as a consequence of the scarce use of electricity during the night. This evidence would suggest preferring the use of HDD and CDD with respect to HCDD.

5.3.4 List of features

In table 5.1 the features used for the forecasting of the three sets of time series IED_d, IED_h and XED_h are reported, where IED_d is composed of a single series, IED_h by 24 and XED_h by 144. The case where the couple HDD and CDD substitutes HCDD is also tested.

Feature	Reference time	Type
Electric demand series	t-1	continuous
Electric demand series	t-7	continuous
Electric demand series	sim(t)	continuous
Electric demand series	sim($t-1$)	continuous
Forecasted temperature	t	continuous
Forecasted temperature	t-1	continuous
Forecasted temperature	t-7	continuous
Forecasted temperature	sim(t)	continuous
Forecasted HCDD	t	continuous
Forecasted HCDD	t-1	continuous
Forecasted HCDD	t-7	continuous
Forecasted HCDD	sim(t)	continuous
Weekday	t	categorical
Holiday	t	dummy
Day after holiday	t	dummy
Bridge holiday	t	dummy

Table 5.1: List of features

5.4 Methodological framework

The same machine learning models, thoroughly described in chapter 3 and tested for the case of GD forecasting in chapter 4, were applied to IED forecasting (IED_d, IED_h and XED_h). These models were structured as follows: nine single models - ridge, LASSO, elastic net, Torus, SVM, GP, KNN, random forest and ANN - followed by five ensemble methods, described in chapter 3, to aggregate the base forecasts - simple average, weighted average, Subset Average (b.f.), Subset Average (c.a.) and SVR aggregation.
Based on a sample of data given by pairs $(\mathbf{x}_i, y_i)$, $i = 1, \dots, n$, where $\mathbf{x}_i \in \mathbb{R}^p$ is the vector of p chosen regressors and y_i is the target variable, in our case respectively IED_d, IED_h and XED_h, each implemented model aims at obtaining a function $y = f(\mathbf{x})$ that forecasts the target variable $y_* = y_*(t_*)$ at a future time t_*, based on the p-dimensional regressor vector $\mathbf{x}_* = \mathbf{x}_*(t_*)$ at that time.
For each model the same hyperparameters and parameters were calibrated as described in chapter 4. In particular, the hyperparameters were chosen through a maximization of the marginal likelihood for the GP, a trial and error procedure followed by a cross validation for the ANN, the AIC minimization for the Torus model and a cross validated grid search for all the other models.
In order to correct the base model predictions, a time series model - SARIMA - was introduced, as described in section 3.1.4, applied to the forecast errors so as to obtain adjusted base forecasts

$$y_*(t_*) = f_i(\mathbf{x}_*) + \epsilon_i^a \tag{5.4}$$

$$= f_i(\mathbf{x}_*) + g_i(\mathbf{x}_*) + \epsilon_i^b \qquad i = 1, ..., 9 \tag{5.5}$$

where $\mathbf{x}_* = \mathbf{x}_*(t_*)$, $f_i(\mathbf{x}_*)$ is the initial non-adjusted base model prediction with error ϵ_i^a, $g_i(\mathbf{x}_*)$ is the error forecast obtained by SARIMA and ϵ_i^b is the i-th base model residual error after SARIMA correction.
Based on the definition of SARIMA model given in section 3.1.4, in this step the time series x_t is given by the base model error ϵ_i^a (eq. (5.4)); the forecast rule $g_i(\mathbf{x}_*)$ of eq. (5.4) is completely defined once the SARIMA hyperparameters and parameters are identified. Following an analysis of autocorrelation of the base model error time series, we highlighted a periodicity with a lag of 7 days in the ϵ_i^a and also some significant 1- and 2-day lag correlation, so that we selected the SARIMA seasonal hyperparameters as $(P, D, Q, s) = (1, 0, 0, 7)$. In particular, as the model errors displayed a cyclostationary behavior, the integral hyperparameter d was fixed equal to zero, while the autoregressive and moving average hyperparameters (p, q) were estimated by minimising the Bayesian information criterion (BIC).
The application of the same model framework - base models, SARIMA correction and aggregation models - was repeated for each of the three target variables.
In case of IED_h and XED_h, the computation times became very long, at least for those of IED_d where the single identification process had to be multiplied 24 and 144 times respectively. In order to reduce the computational burden, the dimensions of the forecast problem were reduced by resorting to PCA, as described in section 3.3.

5.4.1 Hyperparameters

With the exception of the hyperparameters of GP, obtained by marginal likelihood maximization, those of Torus, found by the AIC minimization, and those of ANN, reached by a manual search, all the other model hyperparameters were obtained by a grid search whose grids are reported in table 5.2.

Model	Hyperparameters	Grid
Ridge	alpha	[0.0001, 50], step=50, *log scale*
LASSO	alpha	[0.0001, 50], step=50, *log scale*
Elastic Net	alpha	[0.001, 5], step=10, *log scale*
	l1 ratio	[0.001, 1], step=10
SVM	C	[1000, 10000], step=5
	ϵ	[0.001, 0.01], step=5
	γ	[0.0001, 0.01, 0.1]
KNN	neighbors	[1, 20], step=1
Random Forest	max features	[2, 22], step=5
	max depth	[1, 20], step=1

Table 5.2: Hyperparameters

The KNN was tested both with uniform weights and weights dependent on distance, whereas the kernel of SVM was linear in case of IED_d and squared exponential in case of IED_h and XED_h.

After some trials, the hyperparameters grid was chosen so as to satisfy two conditions: a sufficiently large range for the candidate hyperparameters and a reasonable computation time.

5.5 Experimental framework

As mentioned in section 5.2, the available data range from 2012 to 2018.
The number of test sets were defined on the basis of the different experiments, but all of them spanned over an entire year, from the 1 January to the 31 December. Just as described in chapter 4 for GD forecasting, two training sets were associated to each test sample in order to complete a two-step calibration for the base and ensemble models. First, the base models' parameters were calibrated, from 2012 to the end of the second year before the test set. In this step the base predictors of the last year before the test set were computed as well. Second, the parameters of the ensemble models were calibrated in the last year before the test set. In this step, the base models' parameters were calibrated a second time using the complete training set from 2012 to the last day before the test set. For testing, only models calibrated in the second step were used.
The hyperparameter calibration was automatically executed before each forecasting test. The performance of each of the nine models and five aggregations was measured on each test set using the MAPE:

$$\mathrm{MAPE}_i = \frac{100}{N} \sum_{j=1}^{N} \frac{\left| \mathrm{IED}_j - \hat{\mathrm{IED}}_j \right|}{\mathrm{IED}_j}$$

where $i \in (daily, hourly)$ and N is 365 (or 366) in case of *daily* MAPE and 8760 (or 8784) in case of *hourly* MAPE.
MAPE was chosen rather than MAE due to the almost stationary behavior of IED series. Further, as it is the most frequently used measure of performance in STLF literature we could also compare our results with those published by other authors.
In conclusion, different forecasting experiments were performed, all of them aimed at improving IED forecasting.
Experiment 1. The set of 9 base and 5 ensemble machine learning models was applied, with the error correction afforded by SARIMA to $\mathrm{IED_d}$, obtaining the baseline IED forecasts and MAPEs for the test set years 2015, 2016, 2017, 2018. In this first experiment we used HCDD given by eq. (5.1) where $T_c = 15°\mathrm{C}$ was chosen for maximizing the linear correlation between IED and HCDD.
Experiment 2. As a consequence of the different slopes of the two branches of the U-shape curve representing the relation between IED and temperature, the single HCDD function was replaced with two distinct functions of temperature HDD and CDD. For comparison, the results with this configuration on the same four most recent test years were computed. Then other experiments were performed using a subset of the 9 machine learning models that did not include the Torus and ANN. This choice was made for the sake of simplicity.
Experiment 3. The subset of 7 machine learning models and 5 ensemble ones were applied to the $\mathrm{IED_d}$ forecasting and, based on the similar behavior of each $\mathrm{IED_h}$ with $\mathrm{IED_d}$, to each of the 24 time series of $\mathrm{IED_h}$, using each hour's own hourly temperature forecast as regressor instead of the daily average. After that we could add up the hourly forecasts in

order to obtain the prediction of IED_d. This experiment was defined in order to appreciate the benefit of obtaining daily estimates aggregating a more detailed hourly forecasting.
Experiment 4. The third experiment, where a possible time exploration of the IED_d curve was analysed, was also extended to the spatial dimension. In fact IED is composed by 6 different zones so that it is possible predict each single zone for each of the 24 hours (XED_h). Thus the same machine learning models were applied to each of the 144 XED_h daily curve forecasting problems. In this case as well, the weather temperature regressors were detailed for each zone and hour.
Experiment 5. The 24 and, above all, the 144 times model applications produced better results, at the cost of an increase of the computation time. Therefore, PCA was used to reduce the computation time, another possible benefit being the implicit regularization due to parameter reduction. The results were compared with those of the other experiments in terms of daily and hourly IED MAPEs.

5.6 Results

The results corresponding to the first two experiments are reported in table 5.3 and table 5.4. Both tables show the daily MAPEs of all the tested models pre- and post-correction by SARIMA for the test years from 2015 to 2018, with one difference: in table 5.3 HCDD was used among the regressors, whereas in table 5.4 HDD and CDD were considered.
Some preliminary considerations could be made observing the results in these two tables. The best performing base models are SVM and ANN. On average, SVM is better than ANN probably due to the bad result of ANN in the first test set, caused by the small size of the available train set. In fact the average results for the other three years highlight a better performance by ANN. The same considerations hold before and after the SARIMA correction.
For what concerns the effect of the SARIMA correction, the results are different in terms of level but similar in terms of information content, whether we take into account all the four test years or do not consider year 2015. Indeed, in both cases the most affected models are Random Forest, SVM and KNN, and the average effect of correction on the base models is double with respect to that on aggregated models. On the other hand, in terms of level, if we consider all the four test samples, both for base models and aggregated ones, we measure a value of correction about double with respect to considering only the most recent three years. This result probably is due to the short training set for the test year 2015. All these considerations are shared by results in the two tables 5.3 and 5.4.
The new ensemble model developed for IED forecasting, Subset Average (c.a.), shows, as expected, results similar to Subset Average (b.f.) precisely because both are obtained by a simple average on a optimized subset of predictors which is often the same in both cases.
A further consideration to be made regarding the results in tables 5.3 and 5.4 comes from their comparison and concerns the impact of the temperature-related features. We can conclude that, at daily level, there is no clear evidence about including HCDD rather than the pair HDD and CDD.

	Pre correction				Post correction			
Model	2015	2016	2017	2018	2015	2016	2017	2018
Ridge	2.39	2.17	2.03	2.06	2.17	2.00	1.88	1.93
LASSO	2.37	2.17	2.03	2.06	2.19	2.00	1.88	1.93
Elastic Net	2.40	2.17	2.03	2.06	2.17	2.00	1.88	1.93
SVM	**2.11**	**2.09**	1.85	1.96	**1.99**	**1.96**	1.77	1.86
GP	4.04	2.49	3.15	2.49	3.49	2.44	2.66	2.37
Torus	2.92	2.86	2.48	2.58	2.74	2.66	2.28	2.50
KNN	2.84	2.76	2.48	2.58	2.49	2.39	2.33	2.49
Random Forest	3.05	2.67	2.4	2.41	2.54	2.51	2.20	2.02
ANN	2.52	2.19	**1.68**	**1.66**	2.63	1.97	**1.64**	**1.64**
Simple Average	2.30	1.90	1.80	1.77	2.03	1.76	1.61	1.72
Subset Average (c.a.)	2.51	**1.79**	1.95	1.67	2.3	**1.66**	1.75	1.65
Weighted Average	2.30	1.94	**1.71**	**1.59**	2.03	1.86	**1.59**	**1.59**
SVM aggregation	**2.07**	1.91	1.87	1.62	**2.02**	1.88	1.68	1.72
Subset Average (b.f.)	2.33	1.89	1.91	1.60	2.25	1.79	1.71	1.63

Table 5.3: Daily MAPE on test sets 2015-2018 pre- and post-correction by SARIMA model for IED daily forecasting. Case with HCDD among regressors.

	Pre correction				Post correction			
Model	2015	2016	2017	2018	2015	2016	2017	2018
Ridge	2.50	2.16	2.08	2.07	2.22	1.98	1.90	1.93
LASSO	2.36	2.16	2.08	2.07	2.17	1.98	1.90	1.93
Elastic Net	2.54	2.16	2.08	2.07	2.24	1.98	1.90	1.93
SVM	**2.10**	2.06	**1.87**	1.97	**1.97**	**1.93**	**1.79**	1.86
GP	4.19	2.56	3.11	2.65	3.54	2.58	2.63	2.49
Torus	2.92	2.86	2.48	2.58	2.74	2.66	2.28	2.50
KNN	2.84	2.76	2.48	2.58	2.49	2.39	2.33	2.49
Random Forest	3.13	2.60	2.31	2.33	2.59	2.57	2.01	2.14
ANN	2.48	**2.02**	1.95	**1.77**	2.84	1.95	1.88	**1.73**
Simple Average	2.35	1.86	1.80	1.81	2.04	1.74	**1.64**	1.76
Subset Average (c.a.)	2.44	**1.73**	1.94	1.80	2.38	**1.66**	1.78	1.74
Weighted Average	2.35	1.84	**1.75**	1.72	2.04	1.77	1.66	1.74
SVM aggregation	**2.15**	1.98	1.77	**1.57**	**2.00**	1.89	**1.64**	**1.67**
Subset Average (b.f.)	2.30	1.79	1.92	1.85	2.23	1.76	1.77	1.86

Table 5.4: Daily MAPE on test sets 2015-2018 pre- and post-correction by SARIMA model for IED daily forecasting. Case with HDD and CDD among regressors.

The results relative to experiments 3 and 4 are reported in tables 5.5, 5.6, 5.7 and 5.8. All the results are aggregated and spatially netted at Italian level, in the first two tables temporarily netted at daily level to obtain values of daily MAPEs, whereas in the third and fourth tables the hourly results are not temporarily netted so as to finish with hourly MAPEs.
Regarding the temperature-related features, we reach the same conclusions as before, when HCDD is considered instead of the pair HDD and CDD. This result is clear both when analysing the daily and hourly MAPEs (comparison between table 5.5 and table 5.6 in the case of daily MAPEs and between table 5.7 and table 5.8 for hourly MAPEs).
On the other hand, the results in table 5.5, confirmed by those in table 5.6, highlight the value added given by the hourly simulations with respect to the initial daily ones.

	2017			**2018**		
Model	IED_d	IED_h	XED_h	IED_d	IED_h	XED_h
Ridge	1.88	1.86	1.83	1.93	1.87	1.86
LASSO	1.88	1.87	1.85	1.93	1.85	1.85
Elastic Net	1.88	1.87	1.85	1.93	1.87	1.87
SVM	**1.77**	**1.53**	**1.51**	**1.86**	**1.49**	**1.44**
GP	2.66	2.51	2.51	2.37	2.19	2.18
KNN	2.33	1.91	2.05	2.49	2.03	2.02
Random Forest	2.2	1.64	1.73	2.02	1.71	1.76
Simple Average	**1.68**	1.56	1.6	**1.76**	1.6	1.59
Subset Average (c.a.)	1.87	1.65	1.66	1.84	1.58	1.55
Weighted Average	1.78	**1.49**	**1.49**	1.84	**1.49**	1.44
SVM aggregation	1.69	1.5	1.54	1.99	1.54	**1.42**
Subset Average (b.f.)	1.74	1.54	1.53	1.89	1.54	1.52

Table 5.5: Daily MAPE on test sets 2017-2018 for IED forecasting respectively by daily IED series (IED_d), hourly IED series (IED_h) and 6 hourly XED series (XED_h). Case with HCDD among regressors.

	2017			2018		
Model	IED_d	IED_h	XED_h	IED_d	IED_h	XED_h
Ridge	1.9	1.87	1.83	1.93	1.83	1.87
LASSO	1.9	1.87	1.83	1.93	1.82	1.88
Elastic Net	1.9	1.88	1.84	1.93	1.84	1.89
SVM	**1.79**	**1.51**	**1.53**	**1.86**	**1.45**	**1.44**
GP	2.63	2.53	2.59	2.49	2.28	2.38
KNN	2.33	1.91	2.05	2.49	2.02	2.02
Random Forest	2.01	1.65	1.72	2.14	1.7	1.77
Simple Average	**1.71**	1.57	1.61	**1.77**	1.56	1.6
Subset Average (c.a.)	1.91	1.66	1.68	1.87	1.56	1.59
Weighted Average	1.8	**1.48**	**1.52**	1.87	**1.45**	1.45
SVM aggregation	1.75	1.5	1.58	1.95	1.49	**1.43**
Subset Average (b.f.)	1.91	1.52	1.55	1.89	1.55	1.54

Table 5.6: Daily MAPE on test sets 2017-2018 for IED forecasting respectively by daily IED series (IED_d), hourly IED series (IED_h) and 6 hourly XED series (XED_h). Case with HDD and CDD among regressors.

In these tables, the results show only a negligible advantage given by the more detailed hourly-zonal simulations. Instead, the tables 5.7 and 5.8 do justice to the most detailed simulations. In fact the lowest hourly MAPEs are definitely those associated with XED_h. The disagreement between tables 5.5 and 5.6 is due to the aggregation made from hourly to daily results, which improves the hourly results more or less at the same level of the XED_h results, where a further aggregation of the 6 zones is performed. In order to shed some light on this fact it was necessary to move to a lower level, i.e. hourly MAPE, so that the further zonal aggregation benefit could be appreciated.

	YEAR 2017		YEAR 2018	
Model	IED_h	XED_h	IED_h	XED_h
Ridge	2.29	2.21	2.40	2.35
LASSO	2.31	2.23	2.40	2.34
Elastic Net	2.30	2.24	2.39	2.35
SVM	**1.88**	**1.80**	**1.92**	**1.84**
GP	3.29	3.09	2.88	2.81
KNN	2.64	2.58	2.78	2.62
Random Forest	2.28	2.14	2.40	2.29
Simple Average	1.97	1.93	2.06	2.01
Subset Average (c.a.)	2.07	1.98	2.08	1.99
Weighted Average	**1.85**	**1.78**	**1.92**	**1.83**
SVM aggregation	1.92	1.82	2.01	1.87
Subset Average (b.f.)	1.95	1.87	2.03	1.93

Table 5.7: Hourly MAPE on test sets 2017-2018 for IED forecasting respectively by hourly IED series (IED_h) and 6 hourly XED series (XED_h). Case with HCDD among regressors.

	YEAR 2017		YEAR 2018	
Model	IED_h	XED_h	IED_h	XED_h
Ridge	2.31	2.21	2.38	2.35
LASSO	2.3	2.21	2.38	2.35
Elastic Net	2.32	2.23	2.39	2.36
SVM	**1.88**	**1.82**	**1.93**	**1.86**
GP	3.28	3.23	3.08	3.06
KNN	2.64	2.58	2.79	2.63
Random Forest	2.3	2.14	2.39	2.29
Simple Average	1.98	1.95	2.05	2.02
Subset Average (c.a.)	2.09	2.01	2.11	2.05
Weighted Average	**1.86**	**1.81**	**1.92**	**1.86**
SVM aggregation	1.92	1.87	1.99	1.87
Subset Average (b.f.)	1.94	1.89	2.03	1.96

Table 5.8: Hourly MAPE on test sets 2017-2018 for IED forecasting respectively by hourly IED series (IED_h) and 6 hourly XED series (XED_h). Case with HDD and CDD among regressors.

It is difficult to compare our best results with those reported in [84] and [85] because of some differences: first of all the tested years, but also the different error measure adopted. In fact they use a MAPE based on quarterly-hour data whereas we use a MAPE based on hourly data. Lastly, we do not take much care of holidays' effects, that are not a focal point of our work. Taking them into account could further improve our results, which, nevertheless, seem comparable with those reported in [84],[85].

The last part of this chapter regards the results of *Experiment 5* where the application of PCA was applied to IED forecasting. The PCA was used in order to reduce the dimensions and hence the computation time of the problem for both IED_h and XED_h, that are replaced by IED_h^{PCA} and XED_h^{PCA}. The analysis was executed on 2017 and 2018 data, as test sets, so that it was possible to compare the results with those computed without PCA: IED_d, IED_h and XED_h.
Reducing the number of components entails a new type of error, the reconstruction error, which is given by the approximation of the problem. As the number of components grows, the reconstruction error is reduced but computation times increase as well as the risk of overfitting.
In Tables 5.9, 5.10, 5.11, 5.12 the results using PCA are reported and compared to the previous results without PCA. The first two tables refer to 2017 year as test set, whereas the other two refer to 2018; for each year the first table contains the daily MAPEs and the second one the hourly MAPEs. As reported in these tables, the behaviour of the tested models, as the number of components changes, was different as a consequence of the peculiarities of each model, possibly also because we intentionally did not change the hyperparameter grid between the different experiments. In fact, it was not feasible to explore the hyperparameter grid for each of the 24 or 144 IED forecasting problems.
On the other hand, focusing on the aggregated model results, as the number of components change, the Simple Average often yields the lowest MAPE for IED_h^{PCA} and XED_h^{PCA}. As expected, increasing the number of components reduces the MAPEs for both IED_h^{PCA} and XED_h^{PCA}. Moreover, when the same number of dimensions are chosen, XED_h^{PCA} achieves larger MAPE than IED_h^{PCA} because of its larger reconstruction error due to a larger number of dimensions, i.e. 144 compared to 24.

	IED_h(PCA)						XED_h(PCA)								
Model	n=1	n=3	n=5	n=10	n=15	n=20	n=1	n=3	n=5	n=10	n=15	n=20	IED_d	IED_h	XED_h
Ridge	2.07	1.96	1.91	1.91	1.91	1.91	2.9	1.94	1.91	1.86	1.87	1.88	1.88	1.86	1.83
LASSO	2	1.89	1.83	1.83	1.83	1.83	2.84	1.89	1.86	1.77	1.8	1.82	1.88	1.87	1.85
Elastic Net	2.15	1.88	1.85	1.84	1.84	1.84	2.89	1.83	1.73	1.78	1.83	1.72	1.88	1.87	1.85
SVM	2.32	2.25	2.22	2.21	2.21	2.21	3.35	2.43	2.37	2.28	2.34	2.35	1.77	1.53	1.51
GP	2.31	2.2	2.17	2.17	2.17	2.17	2.97	2.5	2.42	2.41	2.4	2.4	2.66	2.51	2.51
KNN	2.07	1.96	1.91	1.91	1.91	1.91	2.88	1.93	1.9	1.85	1.86	1.87	2.33	1.91	2.05
Random Forest	2.07	1.97	1.92	1.91	1.91	1.91	2.97	1.93	1.9	1.85	1.86	1.87	2.2	1.64	1.73
Simple Average	1.81	1.68	1.63	1.63	1.63	1.63	2.97	1.74	1.67	1.64	1.66	1.64	1.68	1.56	1.6
Simple Average (c.a.)	1.86	1.72	1.66	1.66	1.66	1.66	2.82	1.72	1.65	1.73	1.74	1.62	1.87	1.65	1.66
Weighted Average	1.85	1.74	1.7	1.7	1.7	1.7	2.84	1.77	1.68	1.61	1.64	1.64	1.78	1.49	1.49
SVM aggregation	1.95	1.82	1.76	1.76	1.76	1.76	3.37	1.82	1.77	1.72	1.71	1.71	1.69	1.5	1.54
Simple Average (b.f.)	1.85	1.74	1.69	1.69	1.69	1.69	2.85	1.72	1.65	1.62	1.65	1.62	1.74	1.54	1.53

Table 5.9: 2017 year as test set: daily MAPEs of IED_h and XED_h, using PCA, as the number of components change. In the table also the comparison with MAPEs of IED_d, IED_h and XED_h, without PCA, are represented. All the model results are reported.

	IED_h(PCA)						XED_h(PCA)							
Model	n=1	n=3	n=5	n=10	n=15	n=20	n=1	n=3	n=5	n=10	n=15	n=20	IED_h	XED_h
Ridge	4.4	2.86	2.46	2.33	2.31	2.3	4.71	2.97	2.63	2.36	2.33	2.33	2.29	2.21
LASSO	4.37	2.82	2.39	2.24	2.22	2.22	4.7	2.92	2.58	2.28	2.26	2.26	2.31	2.23
Elastic Net	4.46	2.92	2.48	2.35	2.33	2.33	4.77	2.89	2.54	2.38	2.38	2.28	2.3	2.24
SVM	4.5	3.18	2.81	2.71	2.69	2.69	5.14	3.39	3.1	2.82	2.86	2.87	1.88	1.8
GP	4.65	3.19	2.75	2.63	2.62	2.62	4.87	3.44	3.11	2.91	2.87	2.86	3.29	3.09
KNN	4.4	2.87	2.46	2.33	2.31	2.3	4.79	3	2.66	2.39	2.36	2.35	2.64	2.58
Random Forest	4.4	2.87	2.46	2.33	2.31	2.31	4.87	2.96	2.63	2.35	2.32	2.32	2.28	2.14
Simple Average	4.22	2.66	2.21	2.05	2.03	2.03	4.87	2.79	2.43	2.15	2.12	2.1	1.97	1.93
Simple Average (c.a.)	4.26	2.67	2.23	2.08	2.05	2.05	4.71	2.76	2.4	2.22	2.19	2.06	2.07	1.98
Weighted Average	4.25	2.68	2.23	2.08	2.06	2.05	4.74	2.8	2.44	2.13	2.11	2.11	1.85	1.78
SVM aggregation	4.27	2.7	2.26	2.13	2.11	2.11	5.04	2.79	2.46	2.2	2.14	2.15	1.92	1.82
Simple Average (b.f.)	4.25	2.67	2.23	2.08	2.06	2.05	4.7	2.78	2.44	2.17	2.15	2.12	1.95	1.87

Table 5.10: 2017 year as test set: hourly MAPEs of IED_h and XED_h, using PCA, as the number of components change. In the table also the comparison with MAPEs of IED_h and XED_h, without PCA, are represented. All the model results are reported.

	IED_h(PCA)						XED_h(PCA)								
Model	n=1	n=3	n=5	n=10	n=15	n=20	n=1	n=3	n=5	n=10	n=15	n=20	IED_d	IED_h	XED_h
Ridge	2.07	2	1.98	1.98	1.98	1.98	2.65	2.04	2.04	1.97	1.97	1.97	1.93	1.87	1.86
LASSO	2.07	1.95	1.92	1.92	1.92	1.92	2.63	1.99	2.01	1.94	1.94	1.94	1.93	1.85	1.85
Elastic Net	2.16	2.13	2.1	2.09	2.09	2.04	2.63	2.11	2.17	2.05	2.04	2.05	1.93	1.87	1.87
SVM	2.52	2.34	2.3	2.3	2.3	2.31	2.91	2.4	2.36	2.27	2.26	2.26	1.86	1.49	1.44
GP	2.45	2.32	2.28	2.27	2.27	2.28	3.1	2.51	2.5	2.42	2.41	2.42	2.37	2.19	2.18
KNN	2.07	2	1.97	1.97	1.97	1.97	2.65	2.05	2.09	1.99	1.99	1.99	2.49	2.03	2.02
Random Forest	2.07	2	1.97	1.97	1.97	1.97	2.65	2.03	2.04	1.97	1.97	1.97	2.02	1.71	1.76
Simple Average	1.9	1.79	1.75	1.75	1.75	1.77	2.52	1.89	1.89	1.77	1.77	1.77	1.76	1.6	1.59
Simple Average (c.a.)	1.99	1.88	1.84	1.84	1.84	1.84	2.52	1.99	2	1.82	1.82	1.82	1.84	1.58	1.55
Weighted Average	1.91	1.79	1.75	1.74	1.74	1.79	2.53	1.89	1.9	1.78	1.78	1.78	1.84	1.49	1.44
SVM aggregation	2.09	1.97	1.91	1.91	1.91	1.97	2.62	1.89	1.92	1.92	1.91	1.91	1.99	1.54	1.42
Simple Average (b.f.)	1.9	1.8	1.75	1.75	1.75	1.77	2.55	1.92	1.94	1.81	1.81	1.8	1.89	1.54	1.52

Table 5.11: 2018 year as test set: daily MAPEs of IED_h and XED_h, using PCA, as the number of components change. In the table also the comparison with MAPEs of IED_d, IED_h and XED_h, without PCA, are represented. All the model results are reported.

	IED_h(PCA)						XED_h(PCA)							
Model	n=1	n=3	n=5	n=10	n=15	n=20	n=1	n=3	n=5	n=10	n=15	n=20	IED_h	XED_h
Ridge	4.39	2.88	2.54	2.4	2.38	2.38	4.61	2.95	2.71	2.48	2.43	2.41	2.4	2.35
LASSO	4.4	2.88	2.5	2.35	2.33	2.32	4.61	2.94	2.7	2.46	2.41	2.39	2.4	2.34
Elastic Net	4.37	2.98	2.65	2.53	2.52	2.46	4.59	2.98	2.78	2.58	2.56	2.55	2.39	2.35
SVM	4.56	3.19	2.87	2.75	2.74	2.73	4.76	3.23	2.99	2.78	2.76	2.75	1.92	1.84
GP	4.56	3.13	2.81	2.7	2.68	2.68	4.86	3.29	3.08	2.89	2.85	2.83	2.88	2.81
KNN	4.39	2.88	2.53	2.38	2.37	2.36	4.61	3.01	2.79	2.5	2.45	2.43	2.78	2.62
Random Forest	4.39	2.89	2.53	2.38	2.37	2.36	4.61	2.95	2.71	2.48	2.43	2.41	2.4	2.29
Simple Average	4.25	2.69	2.31	2.16	2.14	2.15	4.51	2.79	2.53	2.28	2.23	2.21	2.06	2.01
Simple Average (c.a.)	4.31	2.76	2.38	2.23	2.22	2.21	4.52	2.84	2.6	2.32	2.28	2.26	2.08	1.99
Weighted Average	4.27	2.71	2.33	2.17	2.15	2.18	4.52	2.81	2.56	2.3	2.24	2.23	1.92	1.83
SVM aggregation	4.31	2.76	2.42	2.28	2.26	2.31	4.53	2.85	2.62	2.41	2.34	2.33	2.01	1.87
Simple Average (b.f.)	4.25	2.71	2.32	2.17	2.16	2.17	4.53	2.81	2.56	2.3	2.25	2.24	2.03	1.93

Table 5.12: 2018 year as test set: hourly MAPEs of IED_h and XED_h, using PCA, as the number of components change. In the table also the comparison with MAPEs of IED_h and XED_h, without PCA, are represented. All the model results are reported.

As reported in fig. 5.7, the daily MAPE analysis of the Simple Average in years 2017 and 2018 highlights that, depending on the year, IED_h^{PCA} and XED_h^{PCA} reach the level of the daily MAPE of IED_d thanks to their higher flexibility, whereas they do not seem to reach the lower levels of IED_h and XED_h, at least for the chosen numbers of components.
Regarding the hourly MAPE, fig. 5.8 displays a similar behavior where IED_h^{PCA} and XED_h^{PCA} approach asymptotically the lower levels of IED_h and XED_h, highlighting the capacity of aggregated models to limit overfitting and increase the robustness of the results.
As previously mentioned, better results in terms of computation time and MAPE could further be obtained by improving the hyperparameter grid of each base model of each component, an option that would be feasible as long as the computational burden does not get too large.

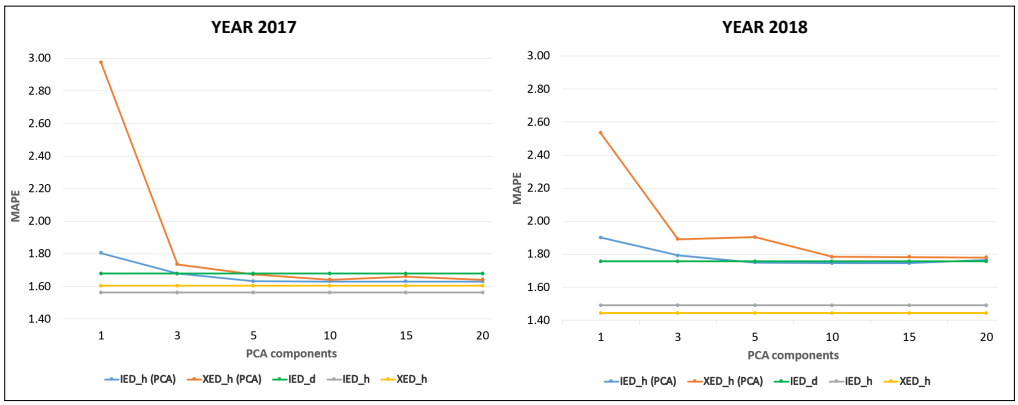


Fig. 5.7: Daily MAPEs of IED_h and XED_h, using PCA, as the number of components change. In the plot also the comparison with MAPEs of IED_d, IED_h and XED_h, without PCA, are represented. Only simple average results are reported.

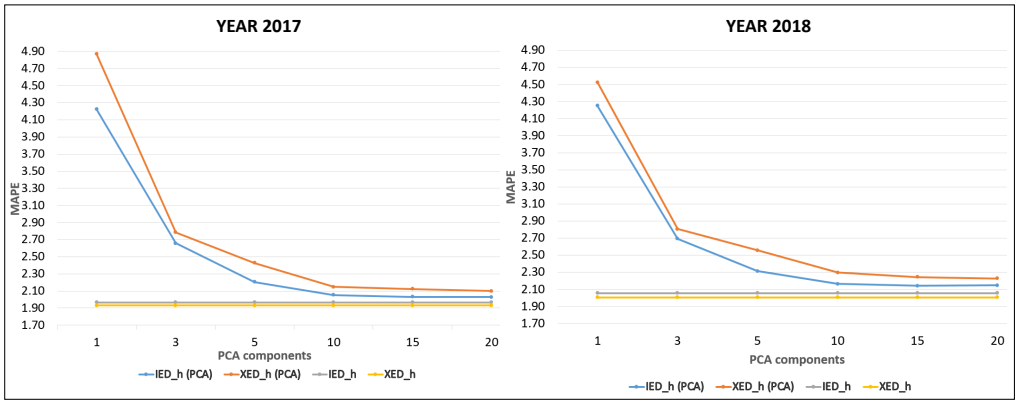


Fig. 5.8: Hourly MAPEs of IED_h and XED_h, using PCA, as the number of components change. In the plot also the comparison with MAPEs of IED_h and XED_h, without PCA, are represented. Only simple average results are reported.

5.7 Conclusion

In this chapter some machine learning techniques for day ahead Italian Electricity Demand forecasting were tested, with prediction errors adjusted by a SARIMA model. We analyzed in depth the effect of different aggregation strategies at model, time and spatial-time level. The model aggregation was obtained by testing 5 different aggregation models tested in different experiments with distinct time (daily and hourly) and spatial-time (Italian and zonal, both at hourly level) aggregation. The results showed lower errors for the aggregation models compared to the base models. Furthermore, the time and spatial-time aggregation strategies highlighted their value leading to proper and better results. We also developed a PCA approach for hourly and zonal-hourly disaggregated series, respectively, in order to cope with the long computation time and the risk of overfitting. Indeed some base models obtain better results with a low number of components, a possible symptom of overparametrization. On the other hand, the aggregation models show a decreasing MAPE as the number of components increases, without reaching the best levels of the models without PCA, at least for the chosen numbers of components. Better forecasts of the Italian electricity demand in terms of MAPE might be achieved by taking account the holidays effect and tuning the hyperparameter grid of each base model of each component, an option that would be feasible as long as the computational burden does not get too large.

6

Recurrent Neural Networks for Italian gas and power demand forecasting

6.1 Introduction and literature review

This chapter is devoted to the application of Recurrent Neural Networks (RNN) to the forecasting of energy time series of gas and electricity demand, so far predicted in chapter 4 and chapter 5 mostly with shallow methods except for the deep MLP (cf. fig. 3.6).
The first RNN architectures were based on the 1986 work [88], where the authors introduced the concept of learning by propagating backward the errors, thereafter widely used in the training methods of different ANN architectures. In that work, the technique was applied to both multi-layer neural networks and recurrent neural networks. The LSTM was introduced in 1997 [29] in order to address the problem of vanishing gradient descent suffered by the plain RNN architecture, whereas the GRU (Gated Recurrent Unit) was proposed in 2014 [33] as a less parametrized alternative to LSTM. All RNN are well suited to process sequences of data, thanks to their memory storing information from the previous elements of the sequence. For this reason, plain RNN, LSTM and GRU have been widely used for handwriting recognition as well as in the field of Natural Language Processing (NLP) for speech recognition, natural language understanding, and natural language generation. In the NLP field, after the first relevant work of Bengio et al. [89], where the authors developed the first neural language model based on feed-forward neural networks, over time the RNN architectures have become the most used [90] with a predominance of the LSTM one [91]. An updated state of the art of the neural language models is given by [92], where the authors compare several modern architectures coming to the conclusion that, often, the plain LSTM architecture outperforms the other more recent models.
Because of their nature, these ANN architectures have been employed for time series forecasting as well. In the energy sector, they have been used more for electricity than for gas. In the most recent literature, plain LSTM architectures have been proposed in [93],[94],[95] in the context of electricity demand forecasting. A plain GRU [96] was used for electricity price forecasting, while alternative architectures which combine CNN with LSTM [97],[98] or some LSTM models on different time scales [99] were also proposed. Two examples of RNN applications to gas demand forecasting are given by LSTM in [48] and GRU in [100]. In their papers, the authors mostly present their network topology and show the results with few details regarding their choice about the different hyperparameters or configurations, except for [95] where the authors compare plain RNN, LSTM, and GRU

architectures with different topologies (layers and units) in order to predict the electric loads and assess the impact of the different configurations.
To the author's knowledge, the procedure to choose the topology and the hyperparameters is seldom explained in detail and also the impact of random initializations is not explored deeply. In this chapter, in order to fill this gap, three plain RNN architectures, a simple RNN (RNN0), an LSTM and a GRU will be studied in connection with the day-ahead predictions of four time series: three Italian gas demands (residential - RGD, industrial - IGD, and thermoelectric - TGD) and the Italian electricity demand (IED). For all these cases, the model identification and the free hyperparameters are obtained by analyzing the MSE loss function in two validation years, after each model has been calibrated in the years before by varying all the combinations of hyperparameters in a grid obtained after some preliminary trials. Then, for some series, a further adjustment is also implemented working out an alternative calibration, in order to face an instability issue highlighted in the first stage of analysis. Particular attention is paid also to the impact of the random initializations in the different cases for all the analyzed models.
This chapter is organized as follows: a description of the experiments carried out in section 6.2, some technical details regarding the Python libraries used in section 6.3, an analysis of the results in section 6.4 and some concluding remarks in section 6.5.

6.2 Experiments

Three RNN architectures, an RNN0, an LSTM, and a GRU, all of them described in chapter 3, were implemented and their forecasting performances compared on the three gas demands and the electricity demand.
The experiments performed are divided into two sections. In the first one, the identification phase, different configurations were studied, tuning the hyperparameters until the best configuration for each of the three models (LSTM, GRU, RNN0) was found. In this phase, the seeds of random number generators were fixed, in order to be able to analyze the effects of model without mixing them with the effect of different random initializations. In the second section, the test phase, the chosen topologies of LSTM, GRU, and RNN0 were applied to the forecasting of Italian gas and electricity demands for the test years 2017-2018; in this phase the seed fixing was relaxed, in order to evaluate the impact of the seed on the final performances and minimize its effect.
We start analyzing the LSTM, which is known as one of the best RNN solution for time series forecasting. Fixing the seed of all the random numbers in a Python session was not a simple problem, because of the different libraries. Below is a piece of code useful to this job

```
import os
import random
import numpy as np
import tensorflow as tf
from keras.initializers import K

def set_all_random_seeds(seed_value = 11):
    # 1. Set `PYTHONHASHSEED` environment variable at a fixed value
    os.environ['PYTHONHASHSEED'] = str(seed_value)
    # 2. Set `python` built-in pseudo-random generator at a fixed value
    random.seed(seed_value)
    # 3. Set `numpy` pseudo-random generator at a fixed value
    np.random.seed(seed_value)
    # 4. Set `tensorflow` pseudo-random generator at a fixed value
    tf.set_random_seed(seed_value)
    # 5. Configure a new global `tensorflow` session
    session_conf = tf.ConfigProto(intra_op_parallelism_threads = 1, inter_op_parallelism_threads = 1)
    sess = tf.Session(graph = tf.get_default_graph(), config = session_conf)
    K.set_session(sess)
```

Fig. 6.1: Function written for fixing the seeds of all the possible random generators. The call to this function ensures the user to obtain the same results after each run of the implemented neural networks models.

Possible LSTM architectures were explored making some trial predictions of the gas demand in a single year, changing some hyperparameters and noting the impact of them on the results, in order to answer two questions: which of them are most relevant and how to build the grid of hyperparameters for their calibration.
After the trials, all the three models and all the series were given: the same batch size (32), the learning rate (0.01), the topology with a single layer, the activation function (Rectified Linear Units - ReLU), the Adam optimizer (with amsgrad=True), the loss function (mean standard error); finally, the hyperparameters' 3×3 grid was given by possible number of units (4, 8, 16) and window length (3, 4, 21).
It was also found that including the HDD features in addition to the raw temperatures was useless because the RNN could directly capture the non-linear relation with the temperature. Given that the model without the HDD features (HCDD in case of IED) achieved equal or better results, the following analyses and tests were performed considering all the features described in chapter 4 except for the four HDD (HCDD in case of IED) related ones.

Model identification

Search of the best topologies for LSTM, GRU, and RNN0.
In order to choose, for each of the three RNN models, the configuration best suited to forecasting RGD, IGD, TGD and IED in the test years 2017 and 2018, we used the years 2015 and 2016 as validation sets. For each of the two validation sets, the nine different configurations were fitted based on previous years. The same procedure was repeated for every one of the three models and the four time-series. The calibration process was run with a considerable number of epochs, 2000, chosen to ensure a high level of fitting and possibly reach the overfitting. The choice of the best configuration for each of the three models and four time-series was obtained analyzing the behavior of the loss measure (MSE) in the two validation sets. The preference was given to those topologies, which showed a behavior with low fluctuations and low level of error. On the other hand, the

number of epochs to use were chosen in order to avoid the growth of the validation loss, caused by the overfitting.

Model testing

Analysis of the forecasts in the test sets.
Once the best configurations had been chosen, the parameters of each model (LSTM, GRU, RNN0) were calibrated for each time series (RGD, IGD, TGD, IED), with all the data preceding the test sets. The results can be appreciated by comparing the levels of MAE for the gas demand forecasts and MAPE for the electricity demand forecasts with those obtained in chapter 4 and chapter 5 by the other machine learning models.

Analysis of alternative forecasting solutions.
In some cases, because of the noisy behavior of the validation loss in the model identification phase, or the unsatisfactory performance registered in the testing section, the adoption of a dynamic learning rate was tried. In particular the Keras callback *ReduceLRonPlateau* [101] was used, which reduces the learning rate of the model when a plateau is reached in the trend of the training loss and, consequently, the model has stopped improving. This Keras callback requires some hyperparameters. Based on the available evidence, we fixed the factor by which the learning rate was to be reduced ($factor$=0.5), the number of epochs with no improvement after which the learning rate had to be reduced ($patience$=50), and the lower bound on the learning rate (min_lr=10^{-5}). For all the other requested hyperparameters, the default values were kept. The Keras callback was added in the model identification stage modifying this process in order to optimize the results: indeed, using *ReduceLRonPlateau*, practically stops the learning process before the end of the process, so that it is possible to fix the number of epochs. The number of epochs was fixed at 500, after some tests which ensured the convergence. The free hyperparameters (units and window) were optimized by cross-validation using all the data before the test set, using years 2017 and 2018 for the final test of the forecasts.

6.3 Technical notes

All the RNN architectures described in this chapter and also the ANN of chapter 4 were implemented in Python 3.6, installed on the operating system Windows 7 by the Anaconda package. Below are the principal libraries used:

Keras==2.1.5
numpy==1.14.2
pandas==0.23.4
pandas-datareader==0.5.0
pandasql==0.7.3
scikit-learn==0.19.0
scipy==1.2.0
SQLAlchemy==1.2.1
statsmodels==0.8.0
tensorflow==1.5.0

tensorflow-tensorboard==1.5.1

Keras was used with the TensorFlow backend, whereas the other machine learning models were implemented by scikit-learn.

6.4 Results

The models were calibrated, by varying the hyperparameters in the grid, table 6.1, using 2015 and 2016 as validation sets.
The validation loss, given by the MSE is displayed as a function of the number of epochs in fig. 6.2 for RGD, fig. 6.4 for IGD, fig. 6.6 for TGD, and fig. 6.8 for IED. All these figures report two rows of three panels; the first raw refers to the validation year 2015 and the second one to 2016, while the three panels refer to LSTM, GRU, and RNN0.
The typical profile of the validation MSE exhibit an initial high level due to the under-fitting, followed by a decrease caused by the learning process, and finally an increase of the MSE attributable to over-fitting. Ideally, one would choose the hyperparameters corresponding to the lowest validation MSE. On the other hand, in real experiments, one should pay due attention to the stability of the behavior of the validation MSE rather than sticking to its exact minimization, which, moreover, could strongly depend by the validation year and the initial point of the optimization process.
Following these criteria, the best configuration was chosen looking at both the MSE curves of the two considered validation years so as to obtain a choice that could prove satisfactory for both the validation sets. The chosen configurations are reported in table 6.2 and the curves of their validation MSE, as a function of the number of epochs, are reported in fig. 6.3 for RGD, fig. 6.5 for IGD, fig. 6.7 for TGD, and fig. 6.9 for IED.

configuration	(window, units)
c1	(3, 4)
c2	(14, 4)
c3	(21, 4)
c4	(3, 8)
c5	(14, 8)
c6	(21, 8)
c7	(3, 16)
c8	(14, 16)
c9	(21, 16)

Table 6.1: Every configurations tested in the model identification phase where the number of epochs is fixed to 2000. Each configuration is defined by the indicated pair (window, units).

The plots of the validation loss for RGD, reported in fig. 6.2, not only show some fluctuations but also, sometimes, an exploding MSE, especially in the case of GRU. On the other hand, as the different curves did not disclose an overfitting behavior, it was possible

to choose a high number of epochs. Summing up, (21, 16, 1800) and (14, 16, 2000) were chosen as the best sets of hyperparameters (window, units, epochs), for LSTM and RNN, respectively. These configurations displayed a stable profile in both the validation years, whereas the one chosen for GRU, (21, 8, 1000), had a volatile behavior in 2016, leading to worse results in the testing phase as reported in table 6.3.

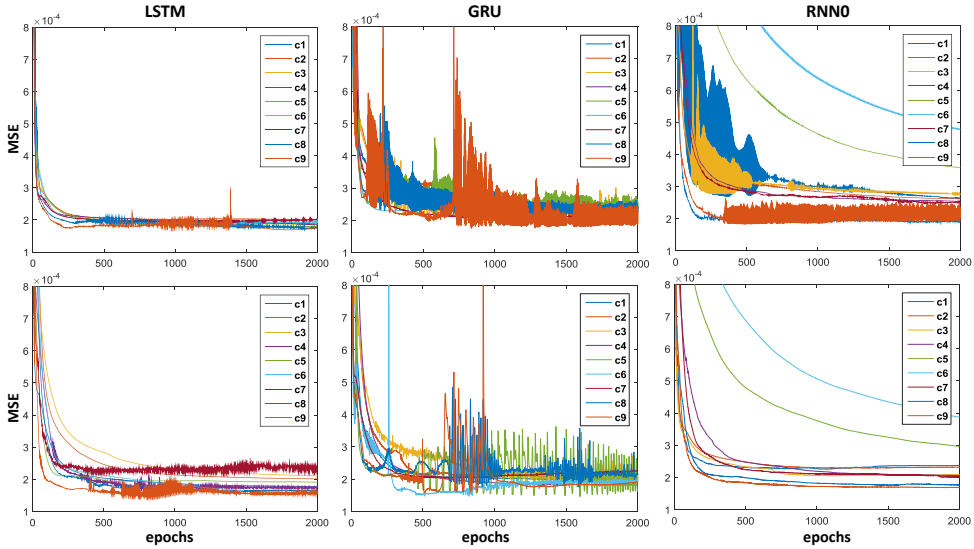


Fig. 6.2: RGD: plot of the validation loss (MSE), as a function of the number of epochs, during the process of model identification for each of the three RNN models and the two validation years 2015 (above) and 2016 (below). In each plot, the nine different configurations of the RNN models are represented, by varying units and window.

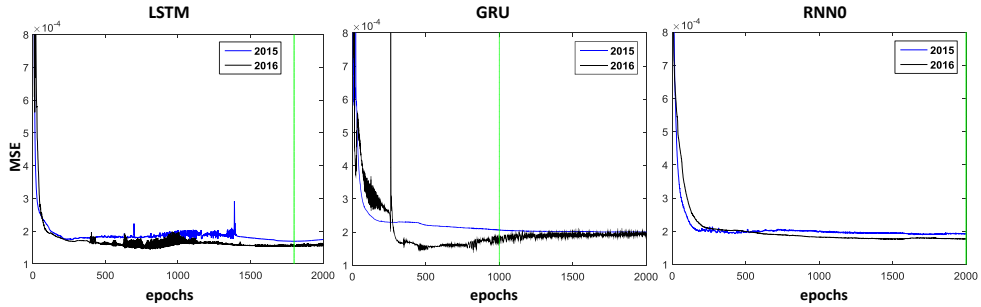


Fig. 6.3: RGD: plot of the validation loss (MSE), as a function of the number of epochs, related to the chosen configurations for each of the three RNN models. The two lines for each box represent the validation loss in the 2015 (blue) and 2016 (black).

The validation loss for IGD series, fig. 6.4, is less noisy but in some cases exhibits clear signs of overfitting, indeed, with the increase of the number of epochs also the MSE grows. However, it was possible to choose steady configurations, as shown in fig. 6.5, and also the maximum number of epochs, 2000, in the cases of LSTM and RNN. The three sets, (window, units, epochs), chosen for the three architectures are reported in table 6.2.

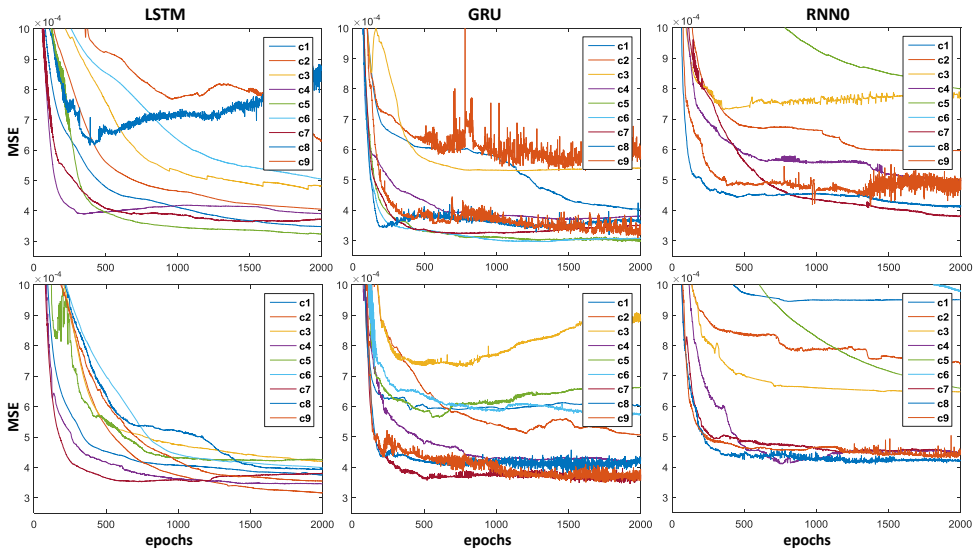


Fig. 6.4: IGD: plot of the validation loss (MSE), as a function of the number of epochs, during the process of model identification for each of the three RNN models and the two validation years 2015 (above) and 2016 (below). In each plot, the nine different configurations of the RNN models are represented, by varying units and window.

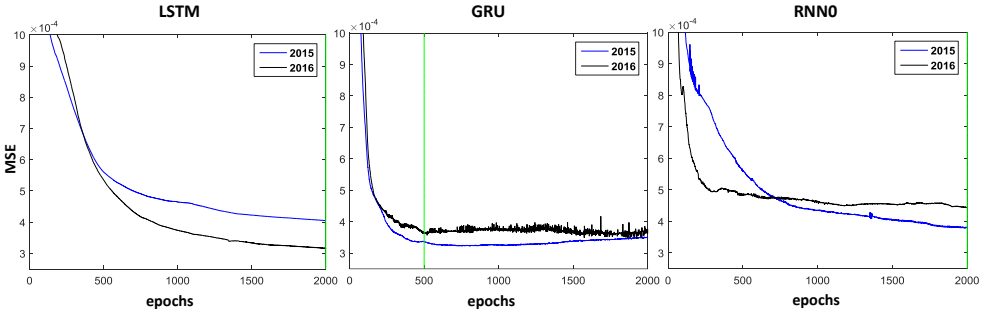


Fig. 6.5: IGD: plot of the validation loss (MSE), as a function of the number of epochs, related to the chosen configurations for each of the three RNN models. The two lines for each box represent the validation loss in the 2015 (blue) and 2016 (black).

In the case of TGD, fig. 6.6, there are many shreds of evidence of overfitting, and this led to choose a smaller number of epochs, 600 for each model. Although some validation losses are noisy, it was possible to choose three configurations, table 6.2, with not very volatile MSE curves, especially close to the chosen epochs (fig. 6.7).

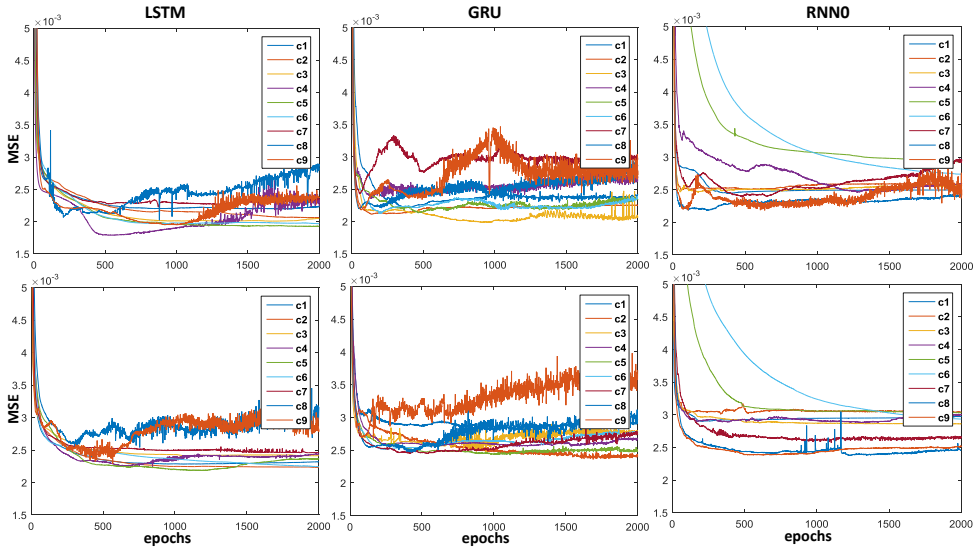


Fig. 6.6: TGD: plot of the validation loss (MSE), as a function of the number of epochs, during the process of model identification for each of the three RNN models and the two validation years 2015 (above) and 2016 (below). In each plot, the nine different configurations of the RNN models are represented, by varying units and window.

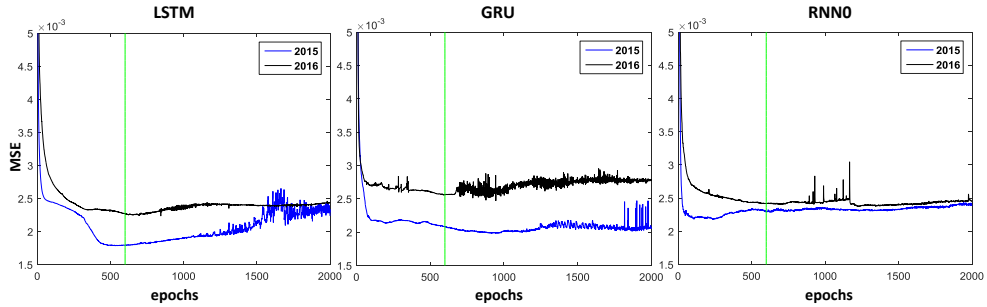


Fig. 6.7: TGD: plot of the validation loss (MSE), as a function of the number of epochs, related to the chosen configurations for each of the three RNN models. The two lines for each box represent the validation loss in the 2015 (blue) and 2016 (black).

In case of IED series, the validation MSE curves exhibit pronounced fluctuations and, in some cases, clear effects due to overfitting which, as for TGD, led to choose a smaller numbers of epochs, 1000 for LSTM and 600 for GRU and RNN0. For the electricity demand, the chosen configuration was the same for the three RNN architectures with window equal to 3 and 8 units. Differently from the other series, in this case, the validation MSE curves for the chosen configurations are noisy, above all for the LSTM. This is a warning regarding the stability of the chosen configurations. This problem will be addressed later with the introduction of the Keras callback ReduceLROnPlateau in the models and a different model identification phase with a different calibration of the hyperparameters.

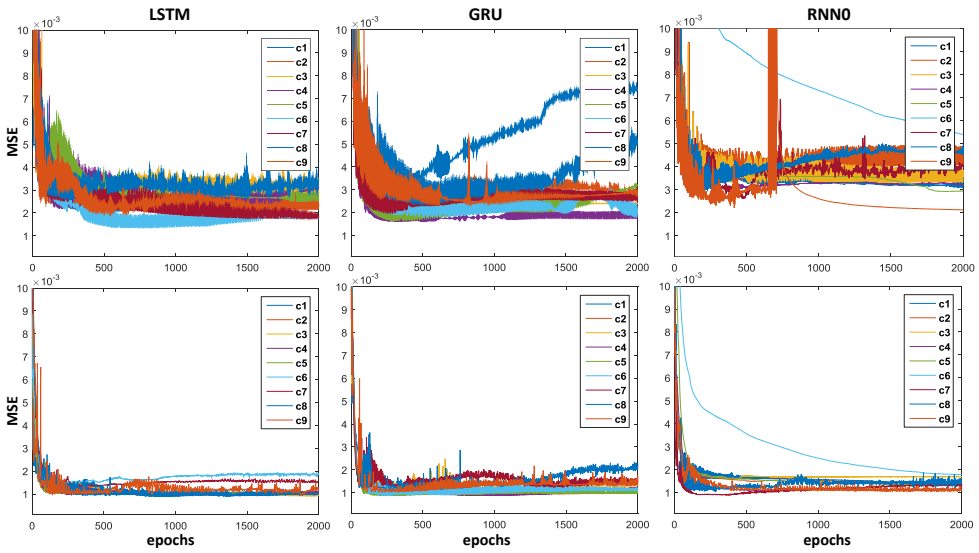


Fig. 6.8: IED: plot of the validation loss (MSE), as a function of the number of epochs, during the process of model identification for each of the three RNN models and the two validation years 2015 (above) and 2016 (below). In each plot, the nine different configurations of the RNN models are represented, by varying units and window.

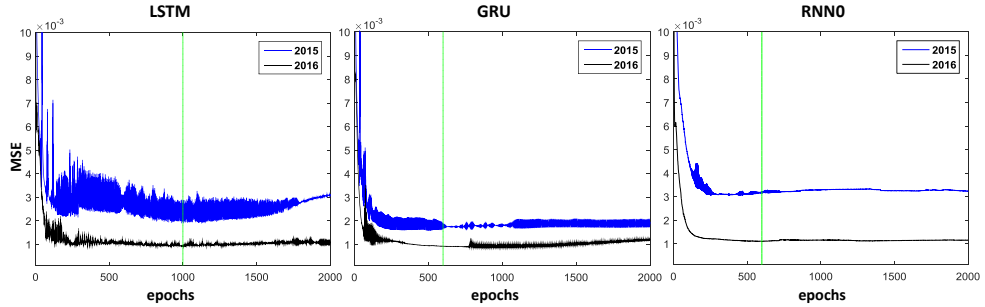


Fig. 6.9: IED: plot of the validation loss (MSE), as a function of the number of epochs, related to the chosen configurations for each of the three RNN models. The two lines for each box represent the validation loss in the 2015 (blue) and 2016 (black).

	LSTM	GRU	RNN0
RGD	(21, 16, 1800)	(21, 8, 1000)	(14, 16, 2000)
IGD	(14, 4, 2000)	(3, 16, 500)	(3, 16, 2000)
TGD	(3, 8, 600)	(21, 4, 600)	(14, 16, 600)
IED	(3, 8, 1000)	(3, 8, 600)	(3, 8, 600)

Table 6.2: The chosen complete configurations (window, units, epochs) for each time-series - model. The complete configurations include also the chosen epochs to use for calibration.

In order to evaluate the forecasting performances of the chosen model configurations taking into account the variability of the forecasts, we computed the forecasts in the test years 2017 and 2018, by changing the initial condition of the optimization process. In these experiments, the seed of the random number generators was not fixed and the operation was replicated ten times so as to obtain ten forecasts for each target variable, model, and test year. The mean values of the 10 forecasts computed for each group (target variable, model, test year), reported in table 6.3, can be compared with the results of RGD (table 4.8), IGD (table 4.9), TGD (table 4.10) and IED (table 5.5) at the same test years. In order to simplify the comparison, the performance indicators (MAE for gas demands and MAPE for electricity demand) of the best base model and the best ensemble model, for the test years 2017 and 2018, are reported in table 6.4. The comparison indicated that the tested RNN models, LSTM, GRU, and RNN0, globally reach similar or better results compared to those previously discussed and shown in table 6.4.
In particular, for the RGD forecasting, the performances of LSTM and RNN0 are both better than those of the best single models, the ANN (MLP) in the 2017 and the GP in the 2018, whereas the GRU reports slightly worse results, probably due to the volatile behavior of its validation loss in the model identification phase.
For the IGD forecasting, all the three RNN architectures show quite similar results between them and better than those of the best single models, for each of the two test sets.
The TGD forecasts obtained by the LSTM are stably better than all the other single model's ones except for that of RNN0 for 2017. On the other hand, the RNN0 forecast in 2018 is worse than that of GRU which shows the worst performances of the three RNN models in 2017.
For what concerns IED forecasting, LSTM, and GRU in the 2017 and GRU in 2018 give the best single model forecasts, also without resorting to any error correction which instead was made, by the SARIMA model, in case of the models presented in chapter 5.
Although the discussed results highlight the good performances achieved by the tested RNN architectures, it is also to point out that it was necessary to compute several forecasts, ten simulations, in order to reduce the variability deriving from the initialization of the optimization process. Indeed, analyzing the results of the single simulations in terms of MAE/MAPE for gas/electricity demand, fairly large values of the mean range (range = maximum-minimum) were observed between the two test years for almost each one of the RNN models and each target variable: 0.71, 0.44, 0.18, 0.49 for the LSTM respectively for RGD, IGD, TGD and IED, 1.11, 0.14, 0.49, 0.64 for the GRU and 0.40, 0.18, 0.41, 0.35 for the RNN0. This result suggests the need of computing an aggregated measure, such as the simple average, in order to obtain more stable results rather than fix the random

seed and compute a single forecast. The best way would be to make many simulations also in the model identification phase, in order to take account of the instability due to the random initialization point also in this initial phase and, consequently, try to reduce it in the choice of the optimal configuration.

	2017			2018		
	LSTM	GRU	RNN0	LSTM	GRU	RNN0
RGD	2.22	2.52	2.12	2.76	3.19	2.80
IGD	0.55	0.57	0.53	0.67	0.69	0.66
TGD	4.17	4.36	4.07	4.32	4.48	4.54
IED	1.48	1.56	1.71	1.71	1.61	1.73

Table 6.3: Performance measures for the three RNN models identified with validation years 2015 and 2016, where also the number of epochs is chosen. Test years 2017 and 2018: daily MAE for the gas demand series and daily MAPE for the electricity demand.

	2017		2018	
	base model	ensemble model	base model	ensemble model
RGD	2.38	2.06	2.83	2.53
IGD	0.57	0.57	0.70	0.61
TGD	4.26	4.17	4.33	4.26
IED	1.77	1.68	1.86	1.76

Table 6.4: Performance measures for the best base model and the best ensemble model in the test years 2017 and 2018: daily MAE for the gas demand series and daily MAPE for the electricity demand.

To stabilize the forecasts of TGD and IED, eventually overcoming the problem observed in the IED hyperparameters calibration, given by the pronounced fluctuations of the validation MSE, first the three models were tested in 2018, adding in the code the Keras callback ReduceLROnPlateau. Then, in view of the unsatisfactory results, the calibration was changed in order to take into account the impact of the coded callback also in the hyperparameters calibration phase. For this purpose, the number of epochs was set to 500, after having verified the achievement of convergence, and the other two hyperparameters were chosen by 5-fold cross-validation based on all the data before the chosen test year. The best configurations (window, units, epochs), for each RNN architecture and each test year, both for TGD and IED, are reported in table 6.5 and their performances, in terms of MAE/MAPE, in the test set 2017 and 2018, are provided in table 6.6.
In the case of TGD, the new method, with the use of the ReduceLROnPlateau callback, renders more comparable the results across the three RNN architectures, improving the GRU's ones and stabilizing those of LSTM and RNN0, but does not improve the results of the best performers. On the other hand, the new method is mostly beneficial to the IED forecasting. In this case, the instability, visible in the calibration phase, was

corrected by the application of the ReduceLRonPlateau callback, with the appropriate hyperparameters calibration and the consequent configuration choice.

	Year	LSTM	GRU	RNN0
TGD	2017	(3, 16, 500)	(3, 8, 500)	(21, 16, 500)
IED	2017	(14, 8, 500)	(3, 8, 500)	(3, 16, 500)
TGD	2018	(3, 4, 500)	(3, 8, 500)	(21, 16, 500)
IED	2018	(3, 8, 500)	(3, 8, 500)	(14, 16, 500)

Table 6.5: Optimal configurations selected by the alternative model identification tested.

	2017			**2018**		
	LSTM	GRU	RNN	LSTM	GRU	RNN
TGD	4.23	4.24	4.15	4.35	4.35	4.41
IED	1.48	1.51	1.54	1.62	1.61	1.68

Table 6.6: Performance measures for the three RNN models identified by cross validation and using the callback *ReduceLRonPlateau*. Test years 2017 and 2018: daily MAE for the gas demand series and daily MAPE for the electricity demand.

6.5 Conclusion

This chapter was devoted to the application of three RNN architectures, LSTM, GRU, and RNN0, to the daily forecasting of the same energy time series, RGD, IGD, TGD, and IED, predicted in the previous chapters by other techniques. In particular, all the results are compared on two annual test sets relative to the most recent years 2017 and 2018, whereas all the previous years were used for model identification, i.e. the tuning of the hyperparameters, and parameter calibration. In addition to the comparison of the RNN model performances with the previous results, the principal focus of this chapter was the methodology followed to choose the best configuration for each RNN architectures, where the sets of free hyperparameters (window, units, epochs) were chosen analyzing the behavior of the MSE loss function related to the epochs in two validation years 2015 and 2016. This analysis highlighted the need for TGD and, above all, for IED, of modifying the models in order to cope with the problem of instability, evidenced by the behavior of MSE in the validation years. The Keras callback *ReduceLRonPlateau* was used in order to reduce the instability, applying a cross-validated grid search for the optimization of the free hyperparameters (window, units), once the epochs were fixed to 500. The forecasts in the two test years were obtained without fixing the random seed, the initial condition of the optimization process at the base of the RNNs calibration, and repeating the computation ten times so as to assess their variance. We compared the simple average

of the ten forecasts, for each target variable, model, and test year, with the previous results.

The results show that RNN models, LSTM, GRU, and RNN0, globally reach similar or better results compared to those discussed in the previous chapters. For the RGD forecasting, LSTM and RNN0 are the best performers, whereas the GRU reports slightly worse results; in the case of IGD forecasting, all the three RNN architectures show quite similar results between them and better than those of the best single models, for each of the two test sets.

In the cases of TGD and IED, the best performer, in each test set, was one of the RNN architectures. Nevertheless, in order to reduce the variability of the performances of the models in the test years, for both series two different calibration procedures had to be used. In these cases, the real value-added, provided by the use of the *ReduceLROnPlateau* callback and the chosen calibration procedure, was represented by the reduction of the variability of the results of the three models, thus easing the choice of the model to use for the prediction.

In spite of the performances obtained by the three RNN architectures, they suffer from a marked instability, due to the dependence on the initialization of the optimization process, however, this randomness was greatly reduced by averaging on ten simulations. A possible improvement could be the use of many simulations, performed without fixing the seed, also in order to identify the optimal model architecture, which would help reducing also this source of variability.

7

Conclusion

In this thesis the day-ahead forecasting of Italian gas demand and Italian electricity loads has been studied through the implementation and testing of nine base models - ridge regression, LASSO, elastic net, Gaussian Process, support vector regression, nearest neighbours, Artificial Neural Networks, torus model, random forest - and five aggregation strategies - simple average, weighted average, support vector regression aggregation and two subset average methods. Moreover, three Recurrent Neural Networks models - LSTM, GRU and simple RNN - were analyzed and tested. All of these methodologies relied on the same set of regressors, properly chosen by means of an exploratory analysis of the time series.
The work provides three main contributions.
First of all, the role of temperature and the propagation of its forecast errors has been put under scrutiny and dissected, especially for gas demand. In fact, the analysis of the time series, and above all of residential gas demand, shows a strong correlation between the demand and the temperature during the cold season which falls to zero during the May-September period, when the temperature is typically higher than 18°C. In order to take advantage of this correlation, it is convenient to include the day-ahead temperature forecast among the regressors. It is then rather natural to ask what is the effect of temperature forecast errors on the final forecast. A first contribution of the thesis was the development of a simple yet accurate formula that quantitatively assess the propagation of temperature forecast errors on the demand prediction. On the Italian data, it was found that the forecast mean square error for the ANN model passes from $\text{RMSE}^2 = 3.64^2 = 13.27$ (using true temperatures, see table 4.2) to $4.03^2 = 16.24$ (using temperature forecasts, see table 4.4). This thing means that temperature forecast errors account for some 18% of the mean square error of gas demand forecasts. As demonstrated on real data, see fig. 4.8, the new error propagation model can successfully predict the quantitative impact of temperature errors on gas forecasts. This could be useful in order to assess the extent and convenience of more sophisticated (and possibly more expensive) weather forecasts.
Second, the performance test of the base predictors, on Italian gas and electricity demand, shows that the best results are achieved by the RNN architectures, LSTM being, on average, the most accurate. Among the other models the best results are achieved by ANN MLP, GP, and SVM in the case of gas series and ANN MLP and SVM in the case of electricity demand. On the other hand, a further improvement is obtained by resorting to so-called ensemble models, which always outperform the base predictors. Other types of

aggregation were tested on the Italian electricity data disaggregated at time and spatial-time levels. As evident in table 5.5, the aggregation of hourly forecasts helps reducing the MAPE with respect to the sole use of daily level data. Analogous considerations apply to the case of table 5.7, where the MAPEs obtained by aggregating forecasts at the zonal level are lower than those associated to national forecasts. Another relevant result is that ensemble models contribute to avoid overfitting, as shown in chapter 5 where the results obtained with the use of PCA are compared with the results without PCA.
A third contribution is the explanation of a possible and practical methodology to identify the optimal configuration of RNN architectures. More precisely, some free hyperparameters (window and units) and the epochs are chosen on the base of the behavior of MSE in two validation years. In some cases, in order to find a remedy to the the instability of the MSE in the validation sets, an alternative method was proposed, based on the dynamical reduction of the learning rate, in order to reduce the volatile behavior. I tested this identification strategy on an LSTM, a GRU and a simple RNN, applied to the forecasting of the four Italian gas and electricity demand series, in the years 2017 and 2018. The accuracy of their performances confirms the viability of the proposed identification methodology.

Notation

Throughout the thesis, boldface is used for vectors and capital boldface for matrices: to provide some examples, the vector of RGD, one of the target variables, is denoted by $\mathbf{y}$, the matrix of the inputs is $\mathbf{X}$ and its i-th column is $\mathbf{x}_i$.
The main symbols and the notation adopted in the thesis are summarised below.

List of Abbreviations

ADAM	Adaptive Moment Estimation
AEEG	Authority of electricity and gas
AIC	Akaike information criterion
ANN	Artificial Neural Networks
ARMA	Autoregressive Moving Average
BIC	Bayesian information criterion
CART	Classification and Regression Trees
CNN	Convolutional neural networks
EUA	European Emission Allowance certificates
FNN	Feedforward neural networks
GD	Italian Gas Demand
GME	Gestore dei Mercati Energetici
GP	Gaussian Process
GRU	Gated Recurrent Unit
IED_d	Total daily IED
IED_h	24-hour components of IED
IED	Italian Electricity Demand
IGD	Industrial Gas Demand
KNN	K-Nearest neighbours
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MGAS	Gas regulated markets
MGPGAS	Day-ahead gas market
MGP	Day-ahead electricity market
MI	Intra-day electricity market
MI-GAS	Intra-day gas market

MLP	Multi-Layer Perceptron
MP-GAS	Spot gas regulated markets
MSD	Ancillary services market
MSE	Mean Squared Error
MT-GAS	Forward gas regulated markets
OTC	Over the counter
PCA	Principal component analysis
PCE	Piattaforma dei conti energia
PSV	Punto di Scambio Virtuale
PUN	Single national price
ReLu	Rectified Linear Unit
RGD	Residential Gas Demand
RKHS	Reproducing Kernel Hilbert Space
RMSE	Root Mean Square Error
RNN0	The simple Recurrent neural network
RNN	Recurrent neural networks
SARIMA	Seasonal Autoregressive Integrated Moving Average
SMP	System marginal price
SRG	Snam Rete Gas
SVM	Support vector machine
TGD	Thermoelectric Gas Demand
TSO	Transmission System Operator
XED_d	Total daily XED
XED_h	24-hour components of XED
XED	Zonal Electricity Demand, for each of the six Italian zones.
CDD	Cooling Degree Day, also indicated as CDD(T) with reference to the temperature T
HCDD	Heating Cooling Degree Day, also indicated as HCDD(T) with reference to the temperature T
HDD	Heating Degree Day, also indicated as HDD(T) with reference to the temperature T
MSCM	Million of Standard Cubic Meter

Mathematical notation

$\mathbb{E}[\cdot]$	Expected value
$\Gamma(\cdot)$	Gamma function
$\kappa(\cdot, \cdot)$	Reproducing kernel
$\|f\|_\kappa$	Squared norm of the function f in the RKHS with reproducing kernel κ
$\mathcal{D} \otimes \mathcal{W}$	Tensor product of two sets $\mathcal{D}$ and $\mathcal{W}$
$\mathcal{H}$	Reproducing Kernel Hilbert Space (RKHS)
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$\odot$	Element-wise product (Hadamard product)
$\text{Var}[\cdot \mid \cdot]$	Conditional variance
$\text{Var}[\cdot]$	Variance
$K_\nu(z)$	Modified Bessel function of the second type with non-negative parameter ν
$p(x \mid y)$	Conditional probability density of x given y

$p(x)$	Probability density function
General forecasting notation	
$(\mathbf{x}_*, y_*)$	Any novel input-output pair
$(\mathbf{x}_i, y_i)$	Training data
$\boldsymbol{\beta}$	Vector of model parameters
$\hat{T}(t)$	One day-ahead forecasted temperature in degrees Celsius at date t
$\mathbf{X}$	Matrix of input features
$\mathbf{y}$	Vector of target observable y
$\mathcal{E}(f)$	Expected risk
$\hat{\mathrm{f}}(\mathbf{x})$	A different way to indicate the one-day-ahead forecast of observable y
$\hat{\mathrm{y}}(t)$	One-day-ahead forecast of observable y at date t
$\mathrm{X}(t)$	Generic observable X at date t
$\sigma(z)$	Sigmoid function
$\tanh(z)$	Hyperbolic tangent function
L^i	Lag operator at lag i
$T(t)$	Temperature in degrees Celsius at date t

References

[1] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[2] Dario Šebalj, Josip Mesarić, and Davor Dujak. Predicting natural gas consumption–a literature review. In *28th International Conference" Central European Conference on Information and Intelligent Systems"*, 2017.

[3] Božidar Soldo. Forecasting natural gas consumption. *Applied Energy*, 92:26–37, 2012.

[4] MS Sachdev, R Billinton, and CA Peterson. Representative bibliography on load forecasting. *IEEE Transactions on Power Apparatus and Systems*, 96(2):697–700, 1977.

[5] Power Systems Engineering Committee Load Forecasting Working Group, A.A. Mahmoud, R.B. Comerford, J Adams, and E Dawson. Load forecast bibliography phase i. *IEEE Transactions on Power Apparatus and Systems*, PAS-99:53–58, 01 1980.

[6] Aly A Mahmoud, Thomas H Ortmeyer, and Robert E Reardon. Load forecasting bibliography phase ii. *IEEE Transactions on Power Apparatus and Systems*, (7):3217–3220, 1981.

[7] Tao Hong et al. Short term electric load forecasting. 2010.

[8] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938, 2016.

[9] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6):594–621, 2010.

[10] Alice Guerini, Andrea Marziali, and Giuseppe De Nicolao. Mcmc calibration of spot-prices models in electricity markets. *Applied Stochastic Models in Business and Industry*, 2019.

[11] Emanuele Fabbiani, Andrea Marziali, and Giuseppe De Nicolao. Short-term forecasting of italian gas demand. *arXiv preprint arXiv:1902.00097*, 2019.

[12] Rafal Weron. *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. Number hsbook0601 in HSC Books. Hugo Steinhaus Center, Wroclaw University of Technology, 2006.

[13] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

[14] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
[15] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
[16] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
[17] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
[18] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.
[19] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York, 2009.
[20] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.
[21] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1, 2000.
[22] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In David Helmbold and Bob Williamson, editors, *Computational Learning Theory*, pages 416–426, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
[23] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. University Press Group Limited, 2006.
[24] K.P. Murphy and F. Bach. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machi. MIT Press, 2012.
[25] Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1965.
[26] Alice Guerini. Long and short term forecasting of daily and quarter-hourly electrical load and price data: a torus-based approach. 2016.
[27] Leo Breiman. *Classification and regression trees*. Routledge, 2017.
[28] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.
[29] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
[30] Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer, 2012.
[31] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.
[32] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.
[33] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[34] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International Conference on Machine Learning*, pages 2342–2350, 2015.

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[36] Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[37] Marek Brabec, Ondřej Konár, Emil Pelikán, and Marek Malỳ. A nonlinear mixed effects model for the prediction of natural gas consumption by individual customers. *International Journal of Forecasting*, 24(4):659–678, 2008.

[38] Primož Potočnik, Marko Thaler, Edvard Govekar, Igor Grabec, and Alojz Poredoš. Forecasting risks of natural gas consumption in slovenia. *Energy policy*, 35(8):4271–4282, 2007.

[39] Salvador Gil and J Deferrari. Generalized model of prediction of natural gas consumption. *Journal of energy resources technology*, 126(2):90–98, 2004.

[40] A Azadeh, SM Asadzadeh, and A Ghanbari. An adaptive network-based fuzzy inference system for short-term natural gas demand estimation: uncertain and complex environments. *Energy Policy*, 38(3):1529–1536, 2010.

[41] Lixing Zhu, MS Li, QH Wu, and L Jiang. Short-term natural gas demand prediction based on support vector regression with false neighbours filtered. *Energy*, 80:428–436, 2015.

[42] Ioannis P Panapakidis and Athanasios S Dagoumas. Day-ahead natural gas demand forecasting based on the combination of wavelet transform and anfis/genetic algorithm/neural network model. *Energy*, 118:231–245, 2017.

[43] Fatih Taşpınar, Numan Celebi, and Nedim Tutkun. Forecasting of daily natural gas consumption on regional basis in turkey using various computational methods. *Energy and Buildings*, 56:23–31, 2013.

[44] Ömer Fahrettin Demirel, Selim Zaim, Ahmet Çalişkan, and Pinar Özuyar. Forecasting natural gas consumption in istanbul using neural networks and multivariate time series methods. *Turkish Journal of Electrical Engineering & Computer Sciences*, 20(5):695–711, 2012.

[45] Jolanta Szoplik. Forecasting of natural gas consumption with artificial neural networks. *Energy*, 85:208–220, 2015.

[46] Božidar Soldo, Primož Potočnik, Goran Šimunović, Tomislav Šarić, and Edvard Govekar. Improving the residential natural gas consumption forecasting models by using solar radiation. *Energy and buildings*, 69:498–506, 2014.

[47] Zlatko Tonković, Marijana Zekić-Sušac, and Marija Somolanji. Predicting natural gas consumption by neural networks. *Tehnički vjesnik*, 16(3):51–61, 2009.

[48] Nan Wei, Changjun Li, Jiehao Duan, Jinyuan Liu, and Fanhua Zeng. Daily natural gas load forecasting based on a hybrid deep learning model. *Energies*, 12(2):218, 2019.

[49] Lorenzo Baldacci, Matteo Golfarelli, Davide Lombardi, and Franco Sami. Natural gas consumption forecasting for anomaly detection. *Expert Systems with Applications*, 62:190 – 201, 2016.

[50] Zia Wadud, Himadri S. Dey, Md. Ashfanoor Kabir, and Shahidul I. Khan. Modeling and forecasting natural gas demand in bangladesh. *Energy Policy*, 39(11):7372 – 7380, 2011. Asian Energy Security.

[51] Yusuf Karadede, Gultekin Ozdemir, and Erdal Aydemir. Breeder hybrid algorithm approach for natural gas demand forecasting model. *Energy*, 141:1269 – 1284, 2017.

[52] Italian natural gas demand report. http://pianodecennale.snamretegas.it/it/domanda-offerta-di-gas-in-italia/domanda-di-gas-naturale.html. Accessed: 2019-01-31.

[53] H Sarak and A Satman. The degree-day method to estimate the residential heating natural gas consumption in turkey: a case study. *Energy*, 28(9):929–939, 2003.

[54] Lifeng Wu, Sifeng Liu, Haijun Chen, and Na Zhang. Using a novel grey system model to forecast natural gas consumption in China. *Mathematical Problems in Engineering*, 2015, 2015.

[55] Bo Zeng and Chuan Li. Forecasting the natural gas demand in China using a self-adapting intelligent grey model. *Energy*, 112:810–825, 2016.

[56] Vincenzo Bianco, Federico Scarpa, and Luca A. Tagliafico. Analysis and future outlook of natural gas consumption in the Italian residential sector. *Energy Conversion and Management*, 87:754–764, nov 2014.

[57] Vincenzo Bianco, Federico Scarpa, and Luca A. Tagliafico. Scenario analysis of nonresidential natural gas consumption in Italy. *Applied Energy*, 113:392–403, jan 2014.

[58] Jon Scott Armstrong. *Principles of forecasting: a handbook for researchers and practitioners*, volume 30. Springer Science & Business Media, 2001.

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

[60] Jakub Nowotarski, Bidong Liu, Rafał Weron, and Tao Hong. Improving short term load forecast accuracy via combining sister forecasts. 2016.

[61] Alice Guerini and Giuseppe De Nicolao. Long-term electric load forecasting: A torus-based approach. In *Control Conference (ECC), 2015 European*, pages 2768–2773. IEEE, 2015.

[62] Snam daily gas consumption forecast. http://www.snam.it/it/trasporto/dati-operativi-business/8_dati_operativi_bilanciamento_sistema. Accessed: 2019-01-31.

[63] Mohamed A Abu-El-Magd and Naresh K Sinha. Short-term load demand modeling and forecasting: a review. *IEEE transactions on systems, man, and cybernetics*, 12(3):370–382, 1982.

[64] George Gross and Francisco D Galiana. Short-term load forecasting. *Proceedings of the IEEE*, 75(12):1558–1573, 1987.

[65] Alex D Papalexopoulos and Timothy C Hesterberg. A regression-based approach to short-term system load forecasting. *IEEE Transactions on Power Systems*, 5(4):1535–1547, 1990.

[66] Takeshi Haida and Shoichi Muto. Regression based peak load forecasting using a transformation technique. *IEEE Transactions on Power Systems*, 9(4):1788–1794, 1994.

[67] O Hyde and PF Hodnett. An adaptable automated procedure for short-term electricity load forecasting. *IEEE Transactions on Power Systems*, 12(1):84–94, 1997.

[68] Slobodan Ruzic, Aca Vuckovic, and Nikola Nikolic. Weather sensitive method for short term load forecasting in electric power utility of serbia. *IEEE Transactions on Power Systems*, 18(4):1581–1586, 2003.

[69] Ibrahim Moghram and Saifur Rahman. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Transactions on power systems*, 4(4):1484–1491, 1989.

[70] James W Taylor, Lilian M De Menezes, and Patrick E McSharry. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22(1):1–16, 2006.

[71] James W Taylor and Patrick E McSharry. Short-term load forecasting methods: An evaluation based on european data. *IEEE Transactions on Power Systems*, 22(4):2213–2219, 2007.

[72] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on power systems*, 16(1):44–55, 2001.

[73] Alex D Papalexopoulos, Shangyou Hao, and Tie-Mao Peng. An implementation of a neural network based load forecasting model for the ems. *IEEE transactions on Power Systems*, 9(4):1956–1962, 1994.

[74] Alireza Khotanzad, Rey-Chue Hwang, Alireza Abaye, and Dominic Maratukulam. An adaptive modular artificial neural network hourly load forecaster and its implementation at electric utilities. *IEEE Transactions on Power Systems*, 10(3):1716–1722, 1995.

[75] Alireza Khotanzad, Malcon H Davis, Alireza Abaye, and D JAMDJ Maratukulam. An artificial neural network hourly temperature forecaster with applications in load forecasting. *IEEE Transactions on Power Systems*, 11(2):870–876, 1996.

[76] Alireza Khotanzad, Reza Afkhami-Rohani, Tsun-Liang Lu, Alireza Abaye, Malcolm Davis, and Dominic J Maratukulam. Annstlf-a neural-network-based electric load forecasting system. *IEEE Transactions on Neural networks*, 8(4):835–846, 1997.

[77] Alireza Khotanzad, Reza Afkhami-Rohani, and Dominic Maratukulam. Annstlf-artificial neural network short-term load forecaster-generation three. *IEEE Transactions on Power Systems*, 13(4):1413–1422, 1998.

[78] Alireza Khotanzad, Enwang Zhou, and Hassan Elragal. A neuro-fuzzy approach to short-term load forecasting in a price-sensitive environment. *IEEE Transactions on Power Systems*, 17(4):1273–1282, 2002.

[79] Kittipong Methaprayoon, Wei-Jen Lee, Sothaya Rasmiddatta, James R Liao, and Richard J Ross. Multistage artificial neural network short-term load forecasting engine with front-end weather forecast. *IEEE Transactions on Industry Applications*, 43(6):1410–1416, 2007.

[80] Nicholas I Sapankevych and Ravi Sankar. Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, 2009.

[81] Shu Fan, Kittipong Methaprayoon, and Wei-Jen Lee. Multiregion load forecasting for system with large geographical area. *IEEE Transactions on Industry Applications*, 45(4):1452–1459, 2009.

[82] M Espinoza, JAK Suykens, R Belmans, and B De Moor. Using kernel-based modeling for nonlinear system identification. *IEEE Control Systems Magazine*, pages 43–57, 2007.

[83] Ramu Ramanathan, Robert Engle, Clive WJ Granger, Farshid Vahid-Araghi, and Casey Brace. Short-run forecasts of electricity loads and peaks. *International journal of forecasting*, 13(2):161–174, 1997.

[84] G De Nicolao, M Pozzi, E Soda, and M Stori. Short-term load forecasting: A power-regression approach. In *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pages 1–6. IEEE, 2014.

[85] Alice Guerini and Giuseppe De 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)*, pages 1–4. IEEE, 2016.

[86] Andrea Marziali, Emanuele Fabbiani, and Giuseppe De Nicolao. Short-term forecasting of Italian residential gas demand. *arXiv e-prints*, page arXiv:1901.02719, January 2019.

[87] Tao Hong et al. Energy forecasting: Past, present, and future. *Foresight: The International Journal of Applied Forecasting*, (32):43–48, 2014.

[88] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[89] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *NIPS*, 2001.

[90] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[91] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[92] Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*, 2017.

[93] Apurva Narayan and Keith W Hipel. Long short term memory networks for short-term electric load forecasting. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2573–2578. IEEE, 2017.

[94] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. Short-term residential load forecasting based on lstm recurrent neural network. *IEEE Transactions on Smart Grid*, 2017.

[95] Sumit Kumar, Lasani Hussain, Sekhar Banarjee, and Motahar Reza. Energy load forecasting using deep learning approach-lstm and gru in spark cluster. In *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pages 1–4. IEEE, 2018.

[96] Umut Ugurlu, Ilkay Oksuz, and Oktay Tas. Electricity price forecasting using recurrent neural networks. *Energies*, 11(5):1255, 2018.

[97] Wan He. Load forecasting via deep neural networks. *Procedia Computer Science*, 122:308–314, 2017.

[98] Chujie Tian, Jian Ma, Chunhong Zhang, and Panpan Zhan. A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network. *Energies*, 11(12):3493, 2018.

[99] Runhai Jiao, Tianming Zhang, Yizhi Jiang, and Hui He. Short-term non-residential load forecasting based on multiple sequences lstm recurrent neural network. *IEEE Access*, 6:59438–59448, 2018.

[100] Primož Potočnik, Jurij Šilc, Gregor Papa, et al. A comparison of models for forecasting the residential natural gas demand of an urban area. *Energy*, 167:511–522, 2019.

[101] François Chollet et al. Keras. https://keras.io, 2015.